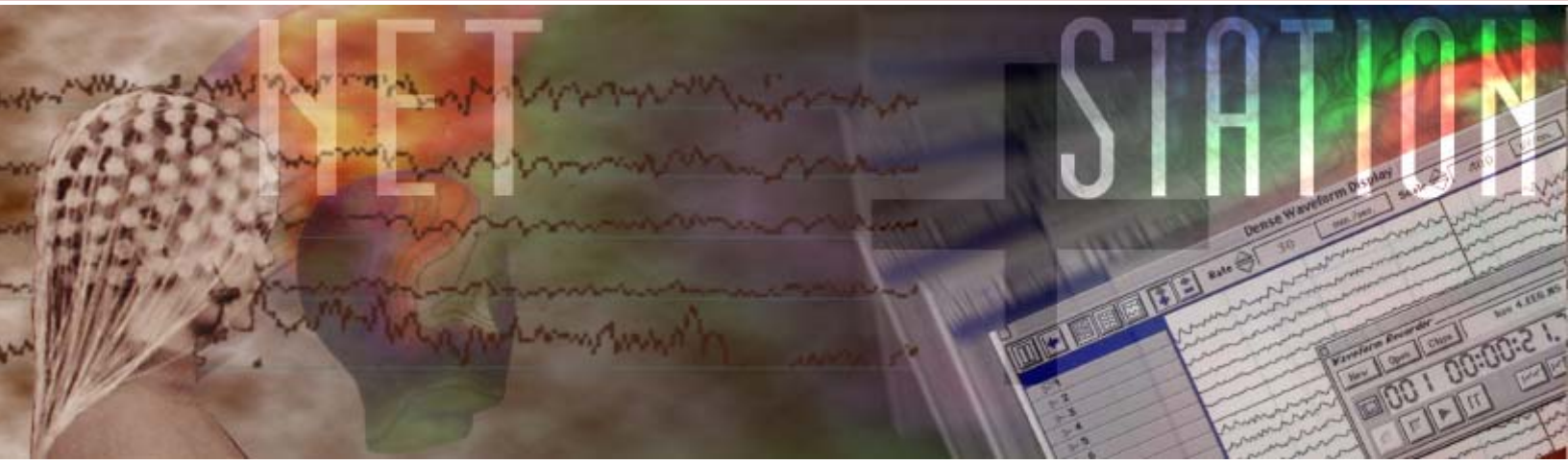
E-Prime Extensions for Net Station



User Manual

Table of Contents

Chapter 1: Getting Started Guide ***4***

E-Prime Extensions for Net Station Overview **4**

Software Installation **4**

Hardware Installation **6**

Running a Test Experiment **14**

Troubleshooting **20**

Technical Support **24**

Chapter 2: Tutorials ***26***

Tutorial 1 – Adding Net Station support to an E-Prime Experiment **26**

Tutorial 2 – Adding user defined keys **54**

Tutorial 3 – Adding a practice block **60**

Chapter 3: Reference Guide ***73***

The Single Clock **73**

The Single Clock timing method *73*

How the Single Clock works *73*

Timing Test **73**

How to review a DIN file with Event Timing Tester *76*

How to review a square wave file *78*

Limitations of the Single Clock *78*

Net Station Package Call Reference **80**

NetStation_Init *82*

NetStation_SendRespEvent *84*

NetStation_SendTrialEvent *86*

NetStation_SendTRSPEvent *87*

NetStation_SendStartRecording *88*

NetStation_StopRecording *89*

NetStation_TrialBegin *90*

NetStation_TrialEnd *91*

NetStation_Uninit *92*

E-Prime Extensions for Net Station User Manual
Table of Contents

Chapter 1: Getting Started Guide

E-Prime Extensions for Net Station Overview

E-Prime Extensions for Net Station is a set of software extensions to E-Prime® that allow communication between Net Station and E-Prime during the run of an experiment. Also included is a set of sample experiments that can be run directly, or used as a basis from which to create new experiments.[1]

Software Installation

National Instruments Driver Installation

Before installing the *E-Prime Extensions for Net Station*, you must install the National Instruments NI-DAQ application, which includes the drivers for the NI DIO96 PCI Data Acquisition card used by the Single Clock hardware. You will install the card later on, in the Hardware Installation section.

Installing the NI-DAQ application:

1. Install the NI-DAQ application from the CD that came with your DIO96 card. You may also download it from the National Instruments website's DAQ Support section, at http://www.ni.com. The current version as of this writing is 7.1.

2. Launch Setup.exe from the CD or NIDAQ 710 folder.

3. Select "Install NI-DAQ/NI-SWITCH" from the list of install options, and wait while the installer initializes.

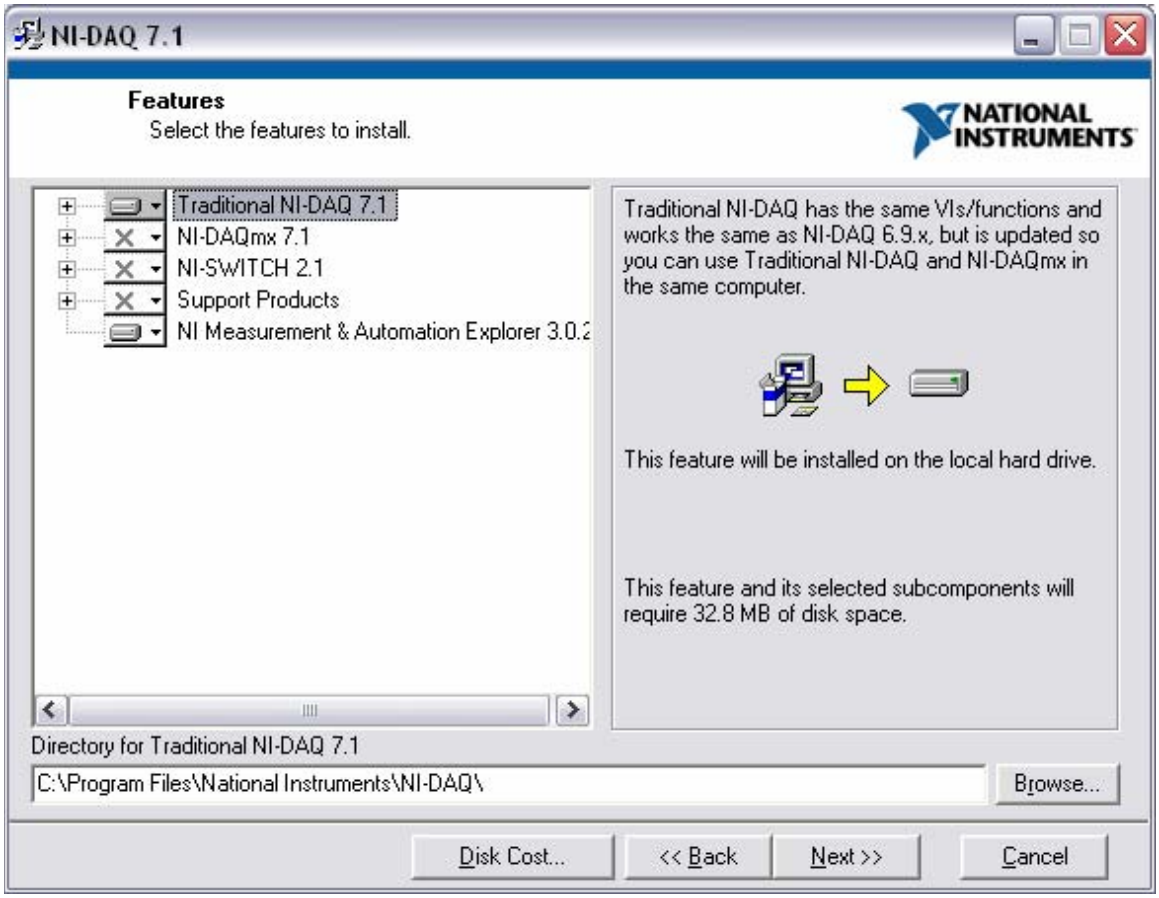
4. Click Next in the following information panel.

5. Click Next again to select the default Destination Directory. This should be: "C:\Program Files\National Instruments\".

6. In the Features list, you may choose to install all (the default) and continue with step 7. However, only "Traditional NI-DAQ 7.1" and "NI Measurement & Automation Explorer 3.0.2" are strictly required.

[1] Please note that this manual is intended for use with the TCP/IP (Ethernet) and Single Clock Experimental Control Interface methods. If you have issues regarding the legacy serial communications or clock scaling methods, please consult the older *E-Prime Biological Add-ons for Net Station User Manual*.

To deselect unnecessary features so that they will not be installed, first click on the icon for "NI-SWITCH 2.1" and choose the option "Do not install this feature" from the pop-up menu. Repeat this for the "NI-DAQmx 7.1" and "Support Products" features.


NI-DAQ 7.1
Features
Select the features to install.
NATIONAL INSTRUMENTS
Traditional NI-DAQ 7.1
NI-DAQmx 7.1
NI-SWITCH 2.1
Support Products
NI Measurement & Automation Explorer 3.0.2
Traditional NI-DAQ has the same VIs/functions and works the same as NI-DAQ 6.9.x, but is updated so you can use Traditional NI-DAQ and NI-DAQmx in the same computer.
This feature will be installed on the local hard drive.
This feature and its selected subcomponents will require 32.8 MB of disk space.
Directory for Traditional NI-DAQ 7.1
C:\Program Files\National Instruments\NI-DAQ\
Browse...
Disk Cost...
<< Back
Next >>
Cancel

7. Click Next in the following three panels, accepting the License Agreements. NI-DAQ 7.1 will now begin installing. This may take a few minutes.

8. When the installation is complete, click Finish and restart your computer.

Installing E-Prime

To install E-Prime follow the installation instructions included with your E-Prime CD. *E-Prime v1.2* or newer is required for use with *E-Prime Extensions for Net Station.*

Installing E-Prime Extensions for Net Station

After installing E-Prime insert the *E-Prime Extensions for Net Station* installation CD into your CD-ROM drive. The SETUP application should automatically launch within 20 seconds. If SETUP does not start automatically, open Windows

Explorer and double-click the SETUP.EXE application in the root directory of the CD.

The installation will update the default E-Prime installation with the files required for use with Net Station[2].

Hardware Installation

TCP/IP ECI

E-Prime and Net Station communicate via Ethernet using a set of standard Internet Protocols. This method is referred to as the *TCP/IP Experimental Control Interface (ECI).* Network traffic may impede ECI communication, so it is strongly recommended that you connect the Macintosh and PC computers directly using a straight-through Category 5 Ethernet cable.

Plug one end of the straight-through Ethernet cable into the Ethernet port on the Net Station Macintosh, and the other end into the Ethernet port on the E-Prime PC.

Single Clock

The Single Clock Experimental Control Timing Box receives a timing pulse from the amplifier and relays a time stamp to the PC, ensuring that the Net Station and E-Prime are temporally synchronized. More information on the Single Clock is provided in the *Single Clock and Timing Test* section of *Chapter 3, Reference Guide*.

To complete the Single Clock hardware installation you will need:

1. A National Instruments DIO96 PCI card.

2. A National Instruments R1005050 Y-shaped, 100 pin to 2 X 50 pin ribbon cable.

3. An EGI Single Clock Experimental Control Timing Box.

[2] This manual assumes that you are installing the software on a computer that will be configured to use the Single Clock hardware. If you are installing the software on a lab machine that will only be used for experiment development, then after the initial installation completes you must manually edit the settings in the E-RUN.INI file to direct the E-Prime runtime to use the default clock (CustomClock="ClockExtension.ebn") rather than the EGI Single Clock (CustomClock ="EgiClockExtension.ebn"). Please refer to the *Troubleshooting* section for more information.

4. An EGI Timing cable. This cable has two variations to match the type of amplifier you are using. EGI can help you select the correct cable for either the Net Amps 200 amplifier, or the Neurotravel USB amplifier.

Install the DIO96 PCI card:

1. Turn off your E-Prime PC and unplug it from the power source.

2. Open it up, and install the DIO96 PCI card in any available PCI slot inside the E-Prime PC. Make sure that the card is seated correctly, and is firmly secured.

3. Close the PC, re-connect it to the power source, and restart it.

4. Once your computer restarts, it will detect the new device and launch the Found New Hardware Wizard. Select “No, not this time” when it prompts you to search for new software, and then click Next.

5. Select “Install the software automatically (Recommended)”, and click Next.

6. Wait while the install completes, and click Finish. You will now see a message that your new hardware has been installed.

Connect the Single Clock Hardware:

1. Connect the 100 pin end of the R1005050 Y-shaped, 100 pin to 2 X 50 pin ribbon cable to the back of the DIO96 PCI card, and tighten the screws to secure it.

2. Connect the 51-100 pin end of the R1005050 cable to the port marked “Experimental Control Computer Digital I/O” on the EGI Single Clock Experimental Control Timing Box. The other end of the cable (pins 1-50) will not be used.

3. Connect the Mini-DIN 5 end of the EGI Timing cable to the port marked “Amplifier Timing Bus” on the EGI Single Clock Experimental Control Timing Box.

4. Connect the other end of the EGI Timing cable to the amplifier.

4a. For the Net Amps 200 amplifier, the EGI Timing cable has a yellow and gray LEMO connector. Plug this into the back of the amplifier in the port marked “Timing: Experimental Synch”, taking care to align the key on the jack with the slot on the connector.

4b. For the Neurotravel USB amplifier, the EGI Timing cable has a 1/8" Mini connector. Plug this into the side of the amplifier in the port marked "Trigger: Out".

Firewall Configuration

If your Macintosh and PC have active local firewalls, you will need to configure them to allow network traffic through the ports that Net Station and E-Prime use to communicate. Opening the relatively obscure ports required for Net Station and E-Prime communication is very safe. You may also choose to disable your firewalls if you are sure you will not re-connect your computers to your network when they are not in use for experiments.

The following instructions are for configuring the firewalls that are built into OS X and Windows XP Professional. The interface for various firewalls is different, but the concept is the same. Please note that for a new installation of OS X, the firewall is off by default. For a new installation of XP SP2, the firewall is on by default but may allow ECI traffic, depending on the current XP security settings. However, it is advisable to set up exceptions for the ECI ports now, to avoid issues later should the firewall configuration change.

Configure the Macintosh OS X Firewall (10.3.7):

1. Launch System Preferences from the Apple menu.

2. Double-click on the Sharing icon in the Internet and Network section.

3. Select the Firewall tab.

4. If your firewall is off, click the Start button.

5. Click the New button to open the port configuration panel.

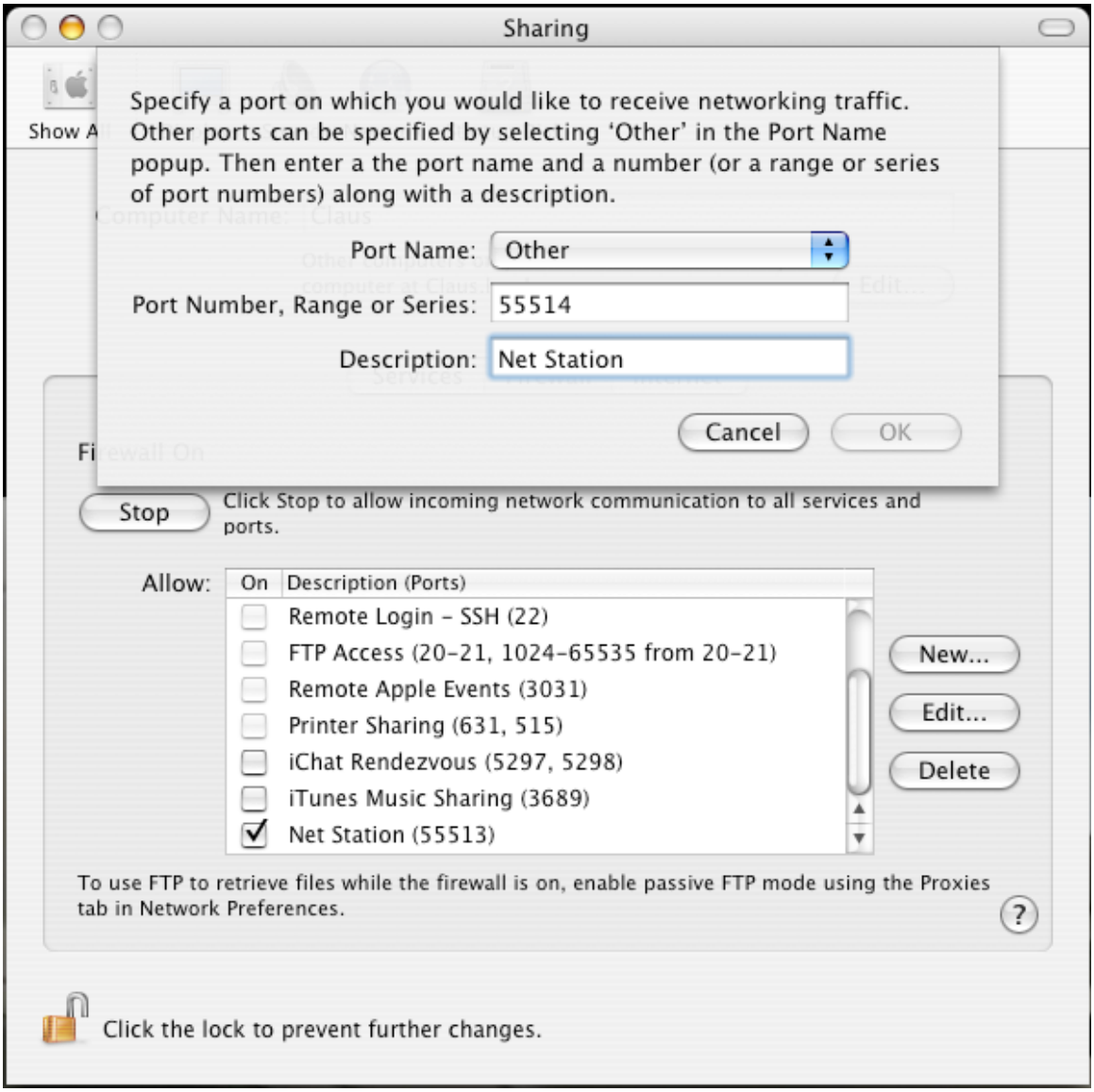
6. Select Other from the Port Name pop up menu.

7. In the "Port Number, Range or Series" text field, type "55513". This is the default port used by Net Station and E-Prime.

8. In the "Description" text field, type "Net Station".

9. Click the OK button to save your port configuration settings.

10. You will now see your new port, “Net Station (55513)” in the “Allow" list. The checkbox for this port will be selected automatically.


Sharing
Specify a port on which you would like to receive networking traffic. Other ports can be specified by selecting ‘Other’ in the Port Name popup. Then enter a the port name and a number (or a range or series of port numbers) along with a description.
Port Name: Other
Port Number, Range or Series: 55514
Description: Net Station
Cancel
OK
Stop
Click Stop to allow incoming network communication to all services and ports.
Allow:
On
Description (Ports)
Remote Login - SSH (22)
FTP Access (20-21, 1024-65535 from 20-21)
Remote Apple Events (3031)
Printer Sharing (631, 515)
iChat Rendezvous (5297, 5298)
iTunes Music Sharing (3689)
Net Station (55513)
New...
Edit...
Delete
To use FTP to retrieve files while the firewall is on, enable passive FTP mode using the Proxies tab in Network Preferences.
Click the lock to prevent further changes.

11. If you want to configure the other three ports that you may use for Net Station/E-Prime communication, repeat steps 5 through 9 for each port. Use port numbers 55514, 55515, and 55516.

12. When you are finished setting up your ports, select “Quit System Preferences” from the main System Preferences menu.

Configure the PC XP Pro Firewall (Windows XP Professional SP2):

1. Open the Windows Firewall control panel by going to Start> Control Panel > Windows Firewall. You may also access it from the Windows Security Center.

2. Select the General tab.

3. If the firewall is off, turn it on by selecting the On radio button.

4. Make sure that the "Don't allow exceptions" checkbox is *not* selected.

5. Select the Exceptions tab.

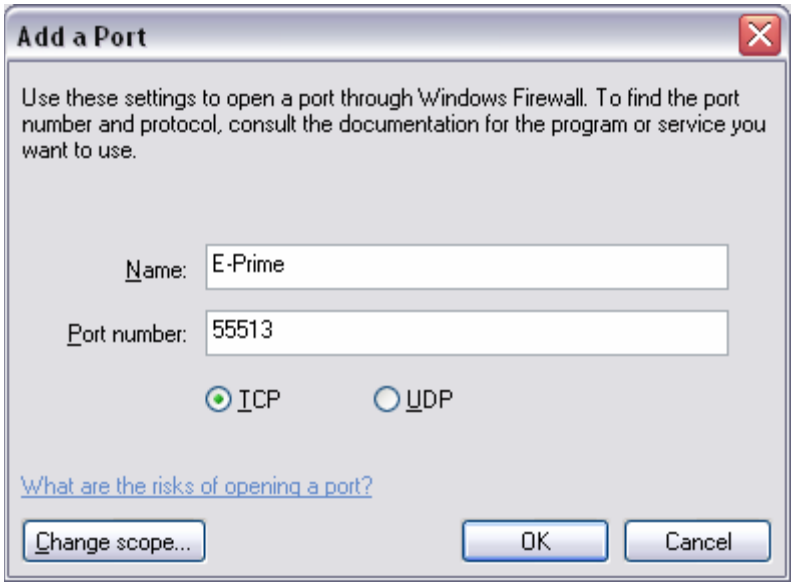
6. Select the "Add Port..." button to open the Add a Port configuration panel.

7. In the "Name" text field, type "E-Prime".

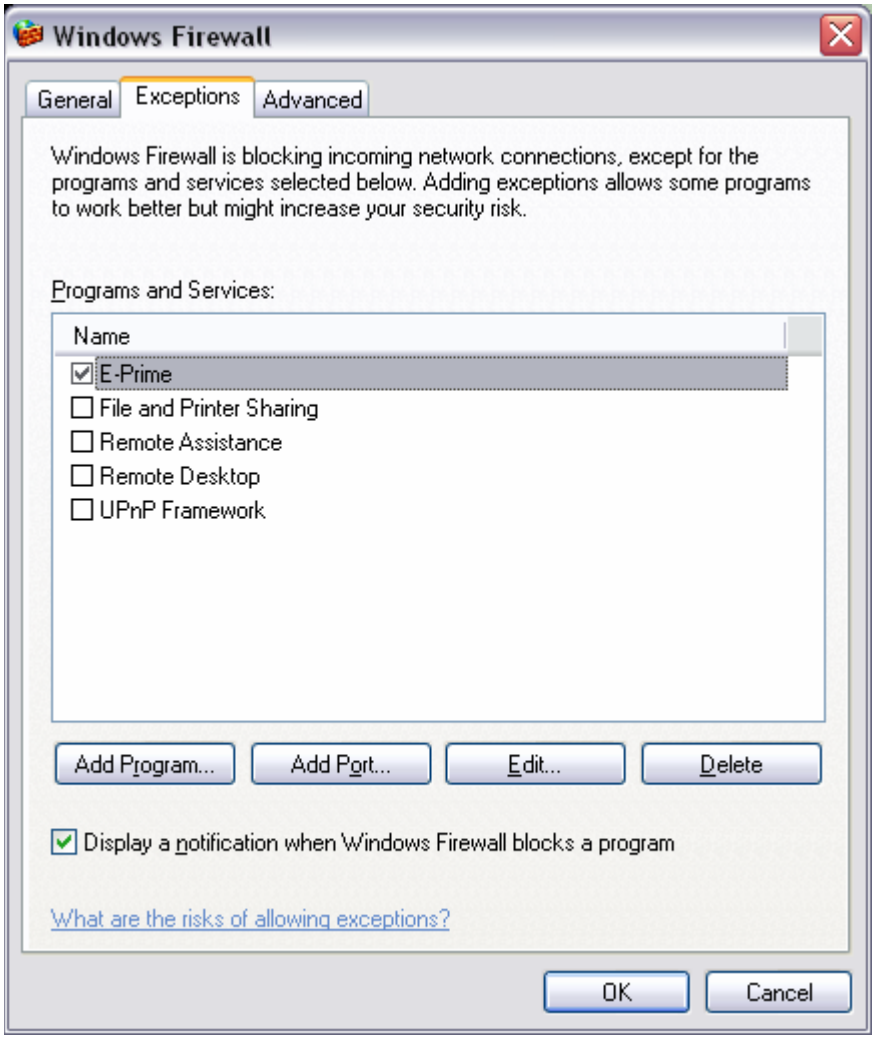
8. In the "Port number:" text field, type "55513".

9. Make sure that the TCP radio button is selected.

10. Click the OK button to save your port configuration settings.


Add a Port
Use these settings to open a port through Windows Firewall. To find the port number and protocol, consult the documentation for the program or service you want to use.
Name:
E-Prime
Port number:
55513
TCP
UDP
What are the risks of opening a port?
Change scope...
OK
Cancel

11. You will now see your new port, "E-Prime" in the "Allow" list. The checkbox for this port will be automatically selected.



12 If you want to configure the other three ports that you may use for E-Prime, repeat steps 6 through 11. Use port numbers 55514, 55515, and 55516.

13. When you are finished setting up your ports, click the OK button to save your new settings and close the Windows Firewall control panel.

TCP/IP Configuration

Because E-Prime and Net Station communicate using TCP/IP, preferably through a straight-through Ethernet cable, you will need to manually set permanent or *static* IP addresses on your PC and Macintosh computers.

DHCP does not retain an IP address once the computer has been disconnected from the network. DHCP may also assign a different IP address to the computer each time it is connected to the network. For these reasons, you must use the Manual IP setting with a static address.

Please contact your Network Administrator if your computers are currently set to DHCP, and you are unsure which static IP addresses to assign to them.

Set a static IP address on your Macintosh (OS 10.3.7):

1. Launch System Preferences from the Apple menu.

2. Double-click on the Network icon.

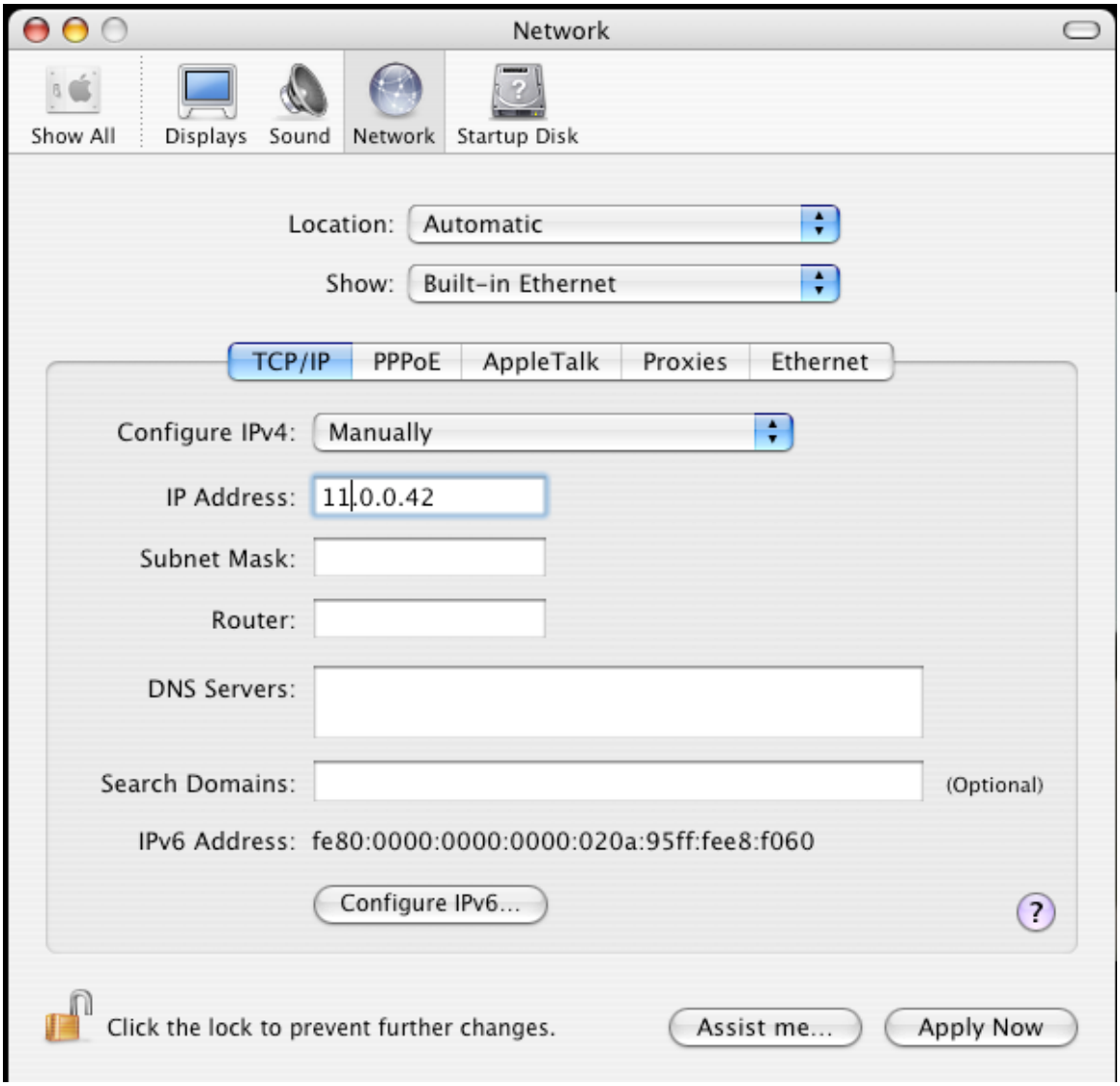
3. Select the current network location from the “Location” pop up menu. This may also be set to “Automatic”.

4. Select “Built-in Ethernet”, from the “Show” pop up menu. If you wish to use a secondary Ethernet device that you have installed, you may select that instead.

5. Select the TCP/IP tab. You will now see your current network settings in panel below.

6. Select “Manually” from the “Configure IP” pop up menu.

7. Type your Macintosh’s static IP address (XXX.XXX.XXX.XXX) into the “IP Address” text box.



8. Click the “Apply Now” button to save your changes, and select “Quit System Preferences” from the System Preferences menu.

Set a static IP address on your PC (Windows XP Professional SP2):

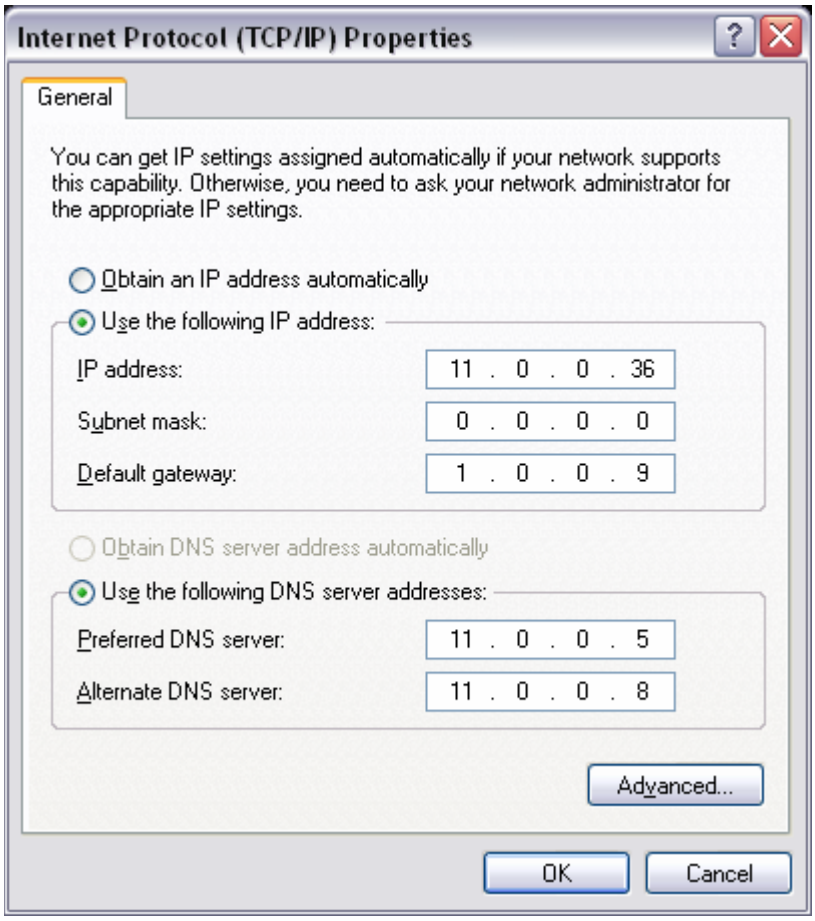
1. Open the Local Area Connection Status control panel by going to Start> Control Panel> Network Connections> Local Area Connection.

2. Click the Properties button to open the Local Area Connection Properties panel.

3. Select “Internet Protocol (TCP/IP)” from the list, and click the Properties button to open the Internet Protocol (TCP/IP) Properties panel.

4. Select the “Use the following IP Address” radio button.

5 Type your PC’s static IP address (XXX.XXX.XXX.XXX) into the “IP address” text field.


Internet Protocol (TCP/IP) Properties
General
You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.
Obtain an IP address automatically
Use the following IP address:
IP address: 11 . 0 . 0 . 36
Subnet mask: 0 . 0 . 0 . 0
Default gateway: 1 . 0 . 0 . 9
Obtain DNS server address automatically
Use the following DNS server addresses:
Preferred DNS server: 11 . 0 . 0 . 5
Alternate DNS server: 11 . 0 . 0 . 8
Advanced...
OK
Cancel

6. Click the OK button to save your selection, and close the Local Area Connection Properties panel.

Running a Test Experiment

Once you have completed the software and hardware installation, and have configured the firewall and TCP/IP settings on your Macintosh and PC computers, you should run a test experiment. In this section, you will use the *NSBasicRT.es* sample experiment provided with the *E-Prime Extensions for Net Station* installer.

This test run will also familiarize you with the procedures for setting up other experiments for TCP/IP ECI communications.

Launch the NSBasicRT.es sample experiment:

1. Launch the E-Studio application from the Windows Start menu by selecting Start> Programs> E-Prime> E-Studio.

2. After E-Studio launches, select “Open an existing Experiment” from the startup dialog.

3. Use the Open File dialog to navigate to appropriate folder, and load the NSBasicRT.es experiment. Under Windows XP, the default file path is:

…\My Documents\My Experiments\Net Station\Samples\NSBasicRT

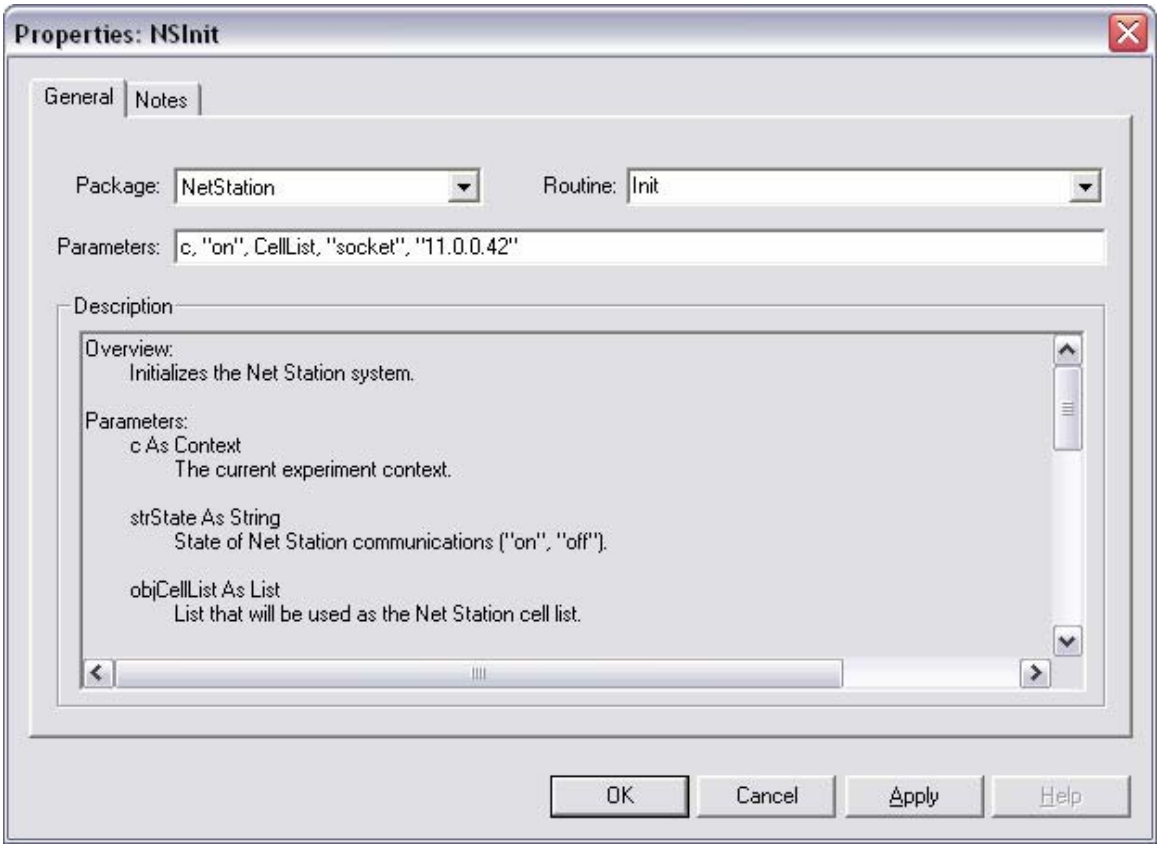
4. You may also launch any E-Studio experiment by double-clicking on the experiment file in Windows Explorer.

Specify the NSInit parameters:

1 Once you have loaded the NSBasicRT.es experiment, you must set it up to communicate using the static IP address of the Net Station Macintosh. Double click on the icon “NSInit” in the Structure View.

2. In the “Parameters” text field, add “socket” and the static IP address (“XXX.XXX.XXX.XXX”) of your Net Station Macintosh. For example, if your IP address was 11.0.0.42, you would enter:

c, “on”, CellList, “socket”, “11.0.0.42”



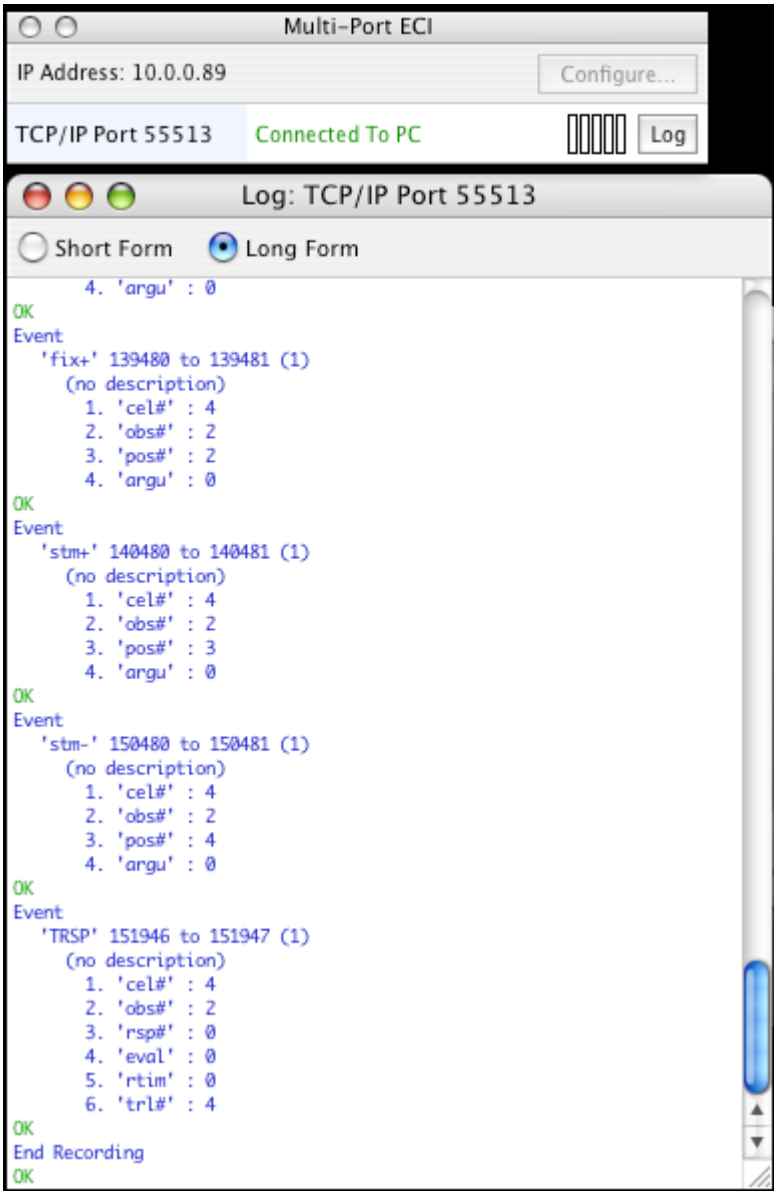
3. Click the OK button to save your changes and close the NSInit object.

Launch Net Station:

1. Follow the instructions provided in your Net Station software manual to create an Acquisition Setup using the Multi-Port ECI Device. Please use the default ECI port, 55513, for this test run.

2. In the Multi-Port ECI panel, click the Log button to open the Log panel.

3. Select the Long Form radio button in the ECI Log panel. This will allow you to view details about events as they come across from E-Prime.



4. Turn on the Workbench to start the amplifier and begin the acquisition session. The Single Clock requires timing from an actively running amplifier.

Run the experiment:

1. Select E-Run> Run from the application menu to generate and run the experiment (alternatively you can press F7, or click the Run button).

2. After the experiment begins executing, it will present message boxes prompting you to enter StartUp Info. Accept the default values or provide a valid value for each the prompt.

3. If the experiment and Net Station have established communications successfully, the experiment will proceed to the experiment instructions and should execute to completion.

During the execution of the experiment trials, Net Station will display the messages being sent from E-Prime within the Multi-Port ECI Log panel and the "ECI TCP/IP 55513" Event Track. It is recommended that you keep these open during testing for reference.

4. If you have trouble running the test experiment, refer to the *Troubleshooting* and *Technical Support* sections at the end of this chapter before continuing.

After you have successfully run NSBasicRT.es and have verified that both systems are working properly, you should proceed to *Chapter 2*, which provides a set of Tutorials that are designed to walk you through the process of adding Net Station support to an existing E-Prime experiment.

Note, prior to collecting any real data with E-Prime and Net Station, it is strongly recommended that you review the critical timing information contained within the Single Clock and Timing Test sections of Chapter 3, Reference Guide.

Alternate NSInit Parameters

1. If you wish to use the TCP/IP Socket ECI method with your legacy Serial ECI experiments, you may want to modify the NSInit Parameters. The example below shows the NSInit Parameters for a typical Serial ECI experiment, NSBasicRT.es:

c, "on", CellList, "serial", "1"

For TCP/IP ECI with the example IP address 11.0.0.42, you would change the NSInit Parameters to:

c, "on", CellList, "socket", "11.0.0.42"

2. E-Prime will use default TCP port 55513 if none is specified in the "Parameters" text field. If you want to use a different port, you may specify it by entering 55514, 55515, or 55516 after the IP address, separated by a colon ("XXX.XXX.XXX.XXX:NNNNN"). For example:

c, "on", CellList, "socket", "11.0.0.42:55514"

3. Net Station will default to port 55513. If you want to use a different port, you many specify it in the Configuration settings of the Net Station Multi-Port ECI panel. You may select multiple ports if you wish, and Net Station will automatically activate the port on which E-Prime sends events during each experimental run.

The NETSTATION.INI file

If you prefer, you may use the NETSTATION.INI file to specify global communication parameters so that there is no need to edit the NSInit package in individual experiments. If no local communication parameters are found in the NSInit package at experiment run time, E-Prime will automatically search NETSTATION.INI for global parameters to use. Local settings in an experiment's NSInit package will always override any global settings in the NETSTATION.INI file.

Please note that you will need an XP User Account with administrative permission in order to edit NETSTATION.INI.

Specify Socket communication parameters in NETSTATION.INI:

1. Navigate to the NETSTATION.INI file. In Windows XP, the default file path is:

    C:\WINDOWS\NETSTATION.INI

2. Right-click on the file and select "Properties". In the General tab, verify that the "Read-only" box is unchecked so that you may edit the file.

3. Under the "[Communications]" header, indicate that you want to use the Socket (TCP/IP) method by entering:

    Method="socket"

4. Now, specify the IP address of you Net Station Macintosh computer ("XXX.XXX.XXX.XXX"). For example:

    MethodParams="11.0.0.42"

5. E-Prime will use default TCP port 55513 if none is specified. If you want to use a different port, you may specify it by entering 55514, 55515, or 55516 after the IP address, separated by a colon ("XXX.XXX.XXX.XXX:NNNNN"). For example:

    MethodParams="11.0.0.0.42:55514"

6. Save and close the NETSTATION.INI file.

Legacy support:

The NETSTATION.INI file also provides support for legacy Serial communication and Clock Scaling methods.

1. Under the "[Timing]" heading, you will see the default entry, "Clockscale=1.0000000000". The Clock scale factor is *never* applied when the Single Clock method is use, so *any value entered there will be ignored*.

2. For Serial communication using Com Port 1: Method="serial" and MethodParams="1". These global Serial parameters are ignored by any experiment with local NSInit Socket parameters.

Troubleshooting

If you have difficulty running an experiment, take note of any error messages that you see and try some of the suggestions below. If you still have issues, please refer to the Technical Support section at the end of this chapter for contact information.

Unable to locate NI-DAQ component:

NSTimingTest.socket.es - E-Studio: E-Studio.exe - Unable To Locate Component
This application has failed to start because nidaq32.dll was not found. Re-installing the application may fix this problem.
OK

If you get a warning at run time that the application failed to start because nidaq.32.dll was not found, check the following:

1. Verify that you have installed the DI-DAQ 7.1 application, which includes the DIO96 drivers.

2. Verify that the Windows XP Device Manager lists the DIO96 card. Check the Properties to verify that the card and drivers are enabled.

3. Reinstall the NI-DAQ 7.1 application.

The clock extension is not found:

1. The “E-Run.ini” file is essential in order for E-Prime to locate the single clock at run time. Verify that E-Run.ini is in the correct file path:

“C:\Program Files\PST\E-Prime\Program\E-Run.ini”.

2. Verify the contents of the “E-Run.ini” file are accurately specified. On a machine using the Single Clock the options should be set to the following by default:

[Options]
CustomClock="EgiClockExtension.ebn"
CustomClockIndex=0

On an experiment development machine (i.e., not connected to or using the Single Clock hardware) the options should be set to the following:

[Options]
CustomClock="ClockExtension.ebn"
CustomClockIndex=0

The clock is not functioning properly:

Runtime Error
The following runtime error occurred:
Unable to communicate with the clock. Please make sure the clock is functioning properly!
Line: -5
Error Number:0
OK

[

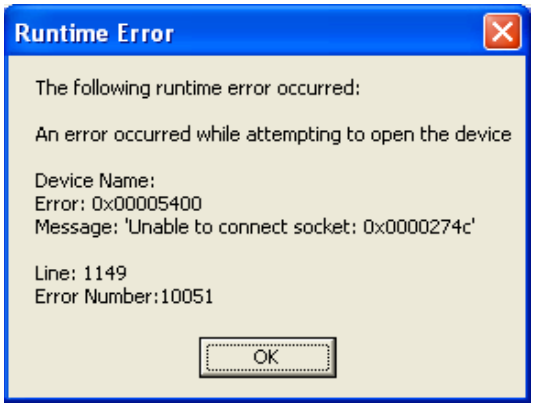
If you get a warning at run time that communication with the clock could not be established, check the following:

1. Verify that you have turned on the Net Station Workbench, the amplifier is on, and data is being displayed in the Net Station Dense Waveform Display before you begin running an experiment. If the amplifier is not actively running, the single clock hardware cannot provide time to E-Prime. Please refer to your Net Station software manual for more assistance with troubleshooting your amplifier and data acquisition setup.

2. Verify that all cables are securely connected to the proper ports on the PC, Single Clock Experimental Control Timing Box, and amplifier.

3. Verify that the DIO96 card and NI-DAQ drivers have been properly installed. You can check the XP Hardware Manager for hardware and driver status.

Communication cannot be verified:



If your E-Prime experiment and Net Station fail to establish TCP/IP (socket) ECI communication check the following:

1. Verify that a straight-through Cat 5 Ethernet cable is connected between the Ethernet ports on the PC and Mac computers. If you are attempting to run over a network connection, try connecting the computers directly instead, to rule out network traffic issues.

2. Review the Parameters of the “NSInit” object in your E-Prime experiment. Refer to the “Running a Test Experiment” section of this chapter for example NSInit Parameters.

3. Verify that the Net Station and E-Prime have the same TCP Port selected.

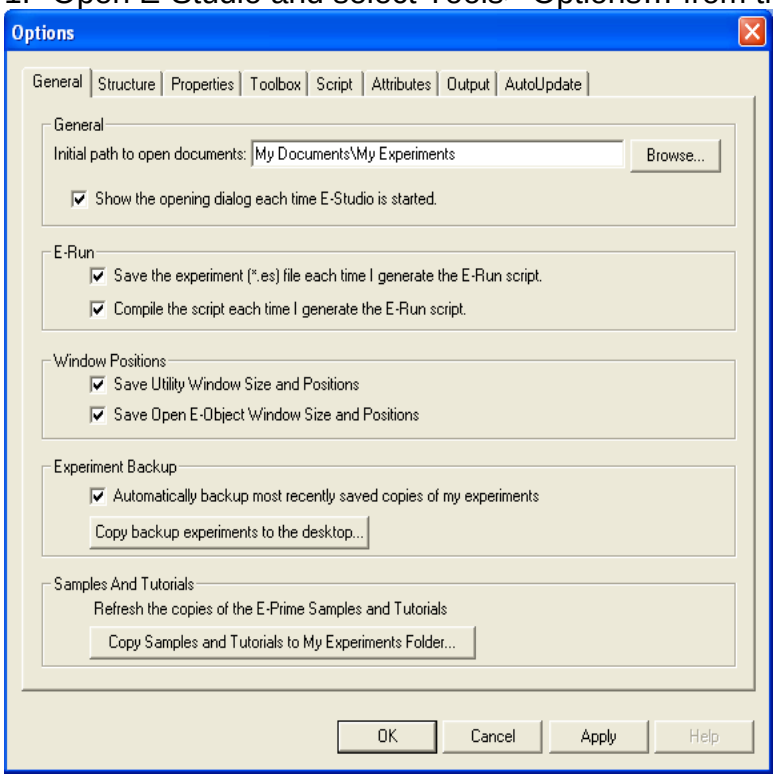
4. Verify that you have set static IP addresses on your PC and Macintosh computers, and that the Network devices and drivers are correctly installed on both machines.

5. If you have active firewalls on your PC and/or Macintosh, verify that you have allowed exceptions for the TCP ports you are trying to use for E-Prime and Net Station ECI communications.

Cannot locate E-Prime or Net Station Sample experiments and Tutorials:

Sample experiments and tutorials should be copied to each users My Documents\My Experiments folder after installation. If you cannot find the Samples and Tutorial experiments or if you just wish to refresh the copies you already have you may force E-Studio to recopy them for you.

1. Open E-Studio and select Tools> Options… from the application menu bar.



2. Click on the “Copy Samples and Tutorials to My Experiments Folder…” button at the bottom of the Options dialog.

3. If there are sample files already there you will be prompted to overwrite them.

Technical Support

Psychology Software Tools, Inc. provides technical support for E-Prime via the PST web site. You must register online at http://www.pstnet.com/e-prime/support/login.asp to receive technical support. To register you simply need a valid serial number. At the support site, you will also find a Knowledge Base including release notes and a compilation of frequently asked questions. In addition, the support site also includes E-Prime sample paradigms that are available for you to download.

Electrical Geodesics, Inc. provides technical support for Net Station and E-Prime Extensions for Net Station. Various support plans are available. E-mail support@egi.com for more information.

Chapter 2: Tutorials

Tutorial 1 – Adding Net Station support to an E-Prime Experiment

Summary: Adding Net Station support to an existing E-Prime experiment primarily consists of the tasks of adding the Net Station Package File to the experiment and then adding PackageCall objects to the experimental structure (at the appropriate locations) in order to communicate information to Net Station. As each PackageCall object is added to the experiment it is typically renamed and edited to set required parameters. These steps will be reviewed in detail.

During this tutorial you will add Net Station support to the PictureRT sample experiment that is installed by default with E-Prime. If you are not familiar with the PictureRT experiment, it is recommended that you load, review and run the sample experiment before beginning the tutorial (e.g. in order to develop an understanding of the overall design of the experiment and how it operates prior to further modifications).

Goal: This tutorial illustrates how to add support for Net Station into the PictureRT sample experiment that is included with E-Prime.

Overview of Tasks:

- Load the PictureRT sample experiment and resave it as NSPictureRT
- Add the Net Station Package File to the Experiment Object
- Enable all of the Startup Info prompts that are required by Net Station
- Add Net Station package calls to initialize/uninitialize the experiment at the Session level
- Add Net Station package calls to start/stop recording at the Block level
- Add Net Station package calls to initialize/uninitialize each trial
- Add Net Station package calls to send RESP, Trial, and TRSP events to Net Station
- Assign unique four character codes to each event of interest within the trial sequence
- Add and configure the CellList
- Assign CellNumbers to individual trials
- Verify the overall experiment structure and run the experiment

Recommended readings: It is recommended that you have read and worked through the tutorials included in the *E-Prime Getting Starting Guide* before beginning this tutorial. Prior to collecting critical data it is recommended that you also read *Chapter 3* of the *E-Prime User's Guide (Critical timing in E-Prime - Theory and recommendations)* as well as the *Single Clock and Timing Test* sections of *Chapter 3, Reference Guide* within this document.

Estimated Tutorial Time: 30-45 minutes.

Task 1: Open the PictureRT.es experiment in E-Studio

Locate the E-Studio icon in the Start>Programs>E-Prime menu and launch the application by selecting it. Load the PictureRT.es sample experiment.

The E-Studio application is installed as part of the standard E-Prime installation. This application is used to create, modify and test experiments within E-Prime. Open the E-Studio application, navigate to the appropriate folder and load the PictureRT.es sample experiment.

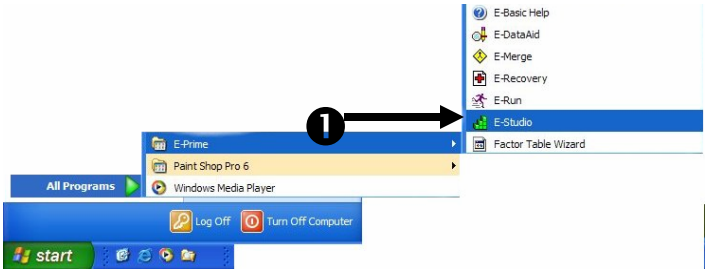
Under Windows XP the folder will be:
"My Documents\My Experiments\Samples\PictureRT"

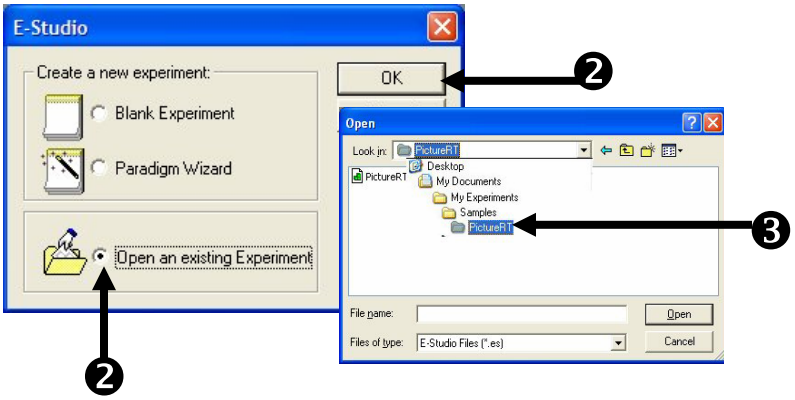
1) ***Click*** on the Windows **Start** menu, ***select*** **Programs**, and then ***select*** **E-Prime**. From the menu, click on **E-Studio** to launch the application.

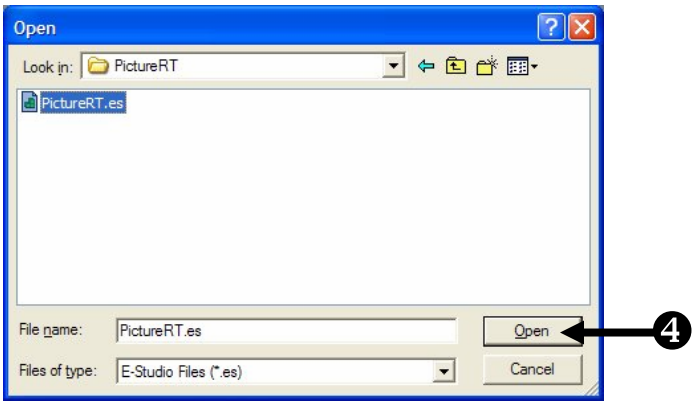
2) ***Select*** **Open an Existing Experiment** and ***click*** the **OK** button.

3) ***Navigate*** to the **"...Samples\PictureRT"** folder to load the paradigm.

4) ***Select*** the **"PictureRT.es"** file and then ***click*** the **Open** button to load the paradigm into E-Studio.







Task 2: Save the experiment under a new name

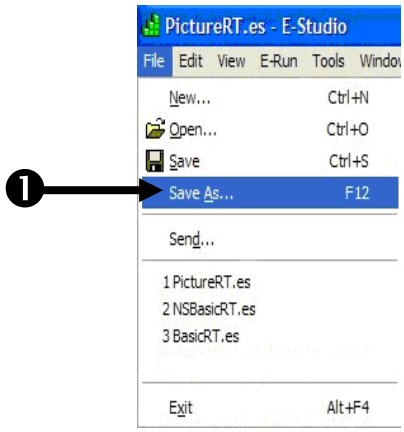
Save the PictureRT.es experiment in the same directory under the new name "NSPictureRT.es".

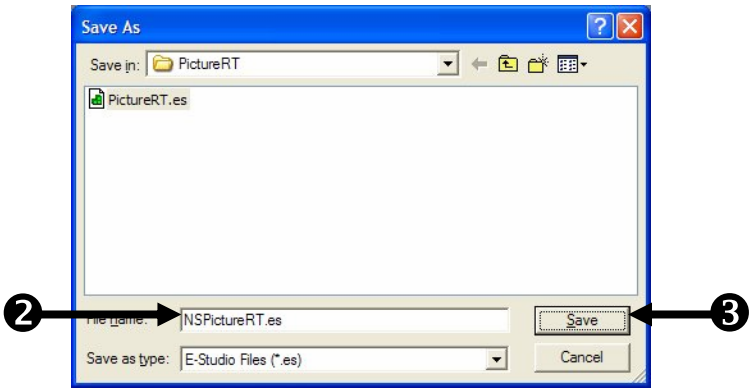
The PictureRT.es sample experiment requires bitmap (i.e. RedCar.bmp, BlueCar.bmp) resources that exist in the current directory. Rename the experiment but be sure to save it in the same directory ("...\Samples\PictureRT") so that the existing bitmap resource references within the experiment will remain valid and can be reused.

1) ***Select*** the **File|Save As...** from the application menu bar.

2) ***Type "NS*PictureRT.es"** as the new name in the **Filename** field.

3) ***Click*** the **Save** button.





Task 3: Add the Net Station Package File to the experiment

Open the Properties dialog for the Experiment Object and use the Packages tab to add the Net Station Package File to the experiment.

Package Files in E-Prime are cohesive sets of E-Basic routines that are grouped together. In order to gain access to the routines within a package file you must first add the package file to the experiment. Package files can be added to an experiment using the Packages tab of the Experiment Object Properties dialog. The routines that are used to communicate with the Net Station hardware and software at runtime are contained within the NetStation Package File.

1) ***Double click*** the **Experiment** object at the top of the tree in the **Structure** view.

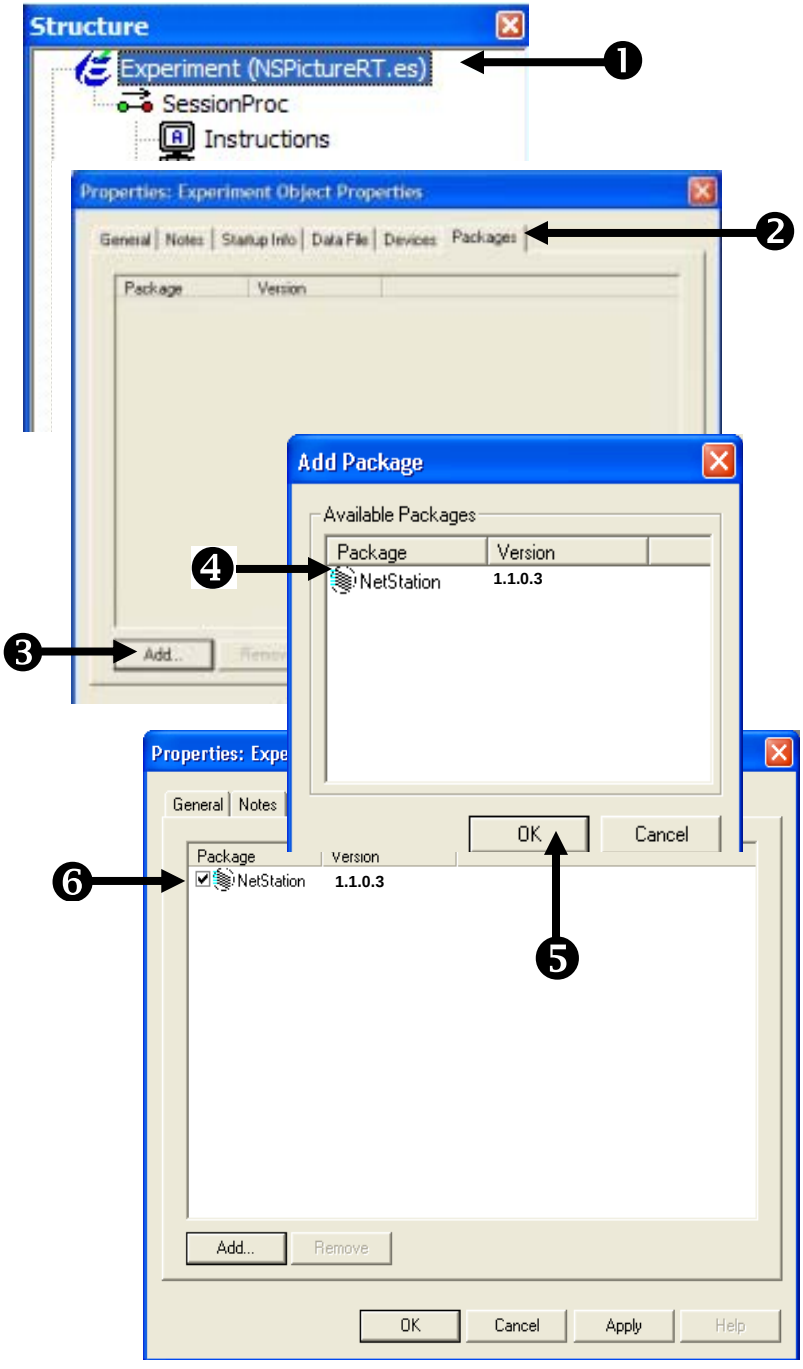
2) ***Click*** on the **Packages** tab of the **Experiment Object Properties** dialog.

3) ***Click*** the **Add...** button.

4) ***Select*** the **NetStation** package file in the **Add Package** dialog.

5) ***Click*** the **OK** button to dismiss the dialog.

6) ***Verify*** the **NetStation** package file has been added and is checked (<u>do not</u> yet click OK or dismiss the dialog).
Note that the Package File version number that is displayed by E-Studio reflects the version of the Net Station Package File that is currently installed on your machine and may not match the picture.



Task 4: Configure the experiment Startup Info

Configure the experiment to prompt for the Startup Info parameters that are required by Net Station.

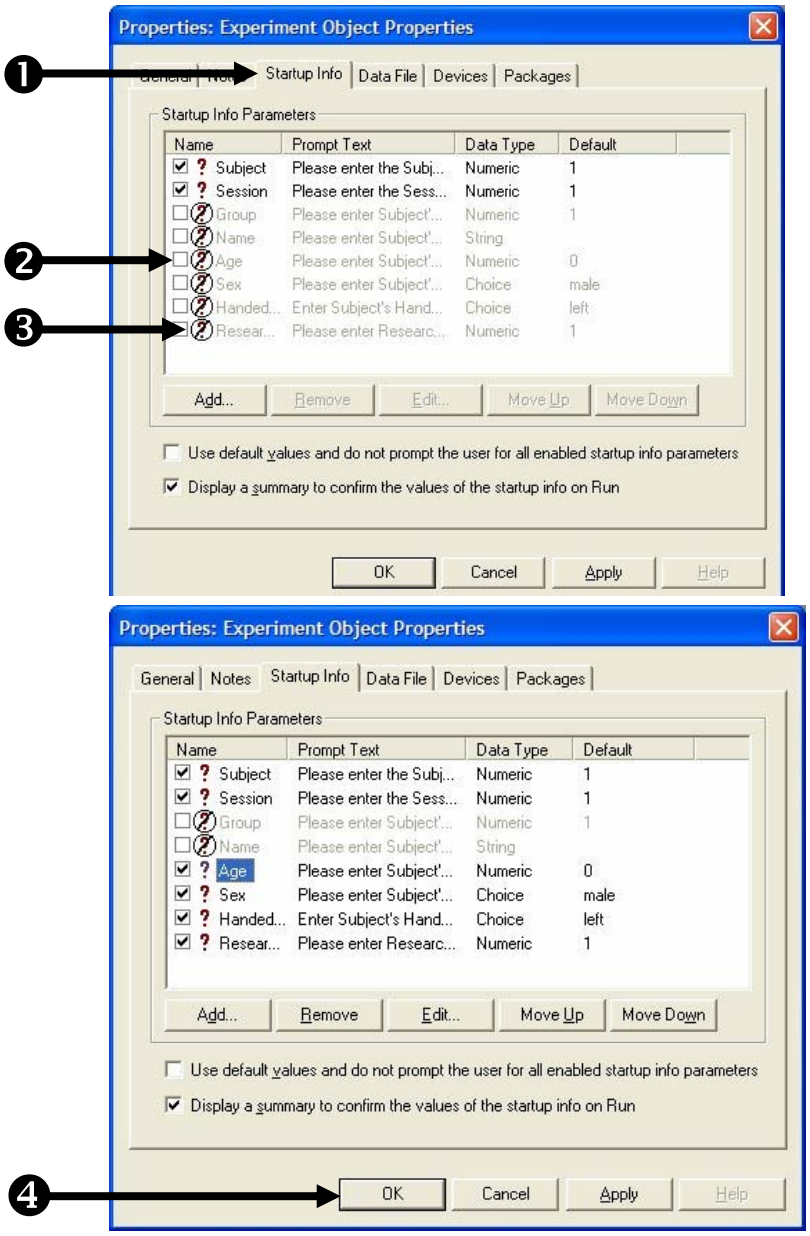
Net Station requires that experiments provide information related to Subject, Session, Age, Sex, Handedness, and ResearcherID. Subject and Session are already enabled in E-Prime by default, but all of the other parameters must be added to the experiment using the Startup Info tab of the Experiment object Properties dialog. It is typical to prompt the researcher for this information at the beginning of the experiment. If you do not prompt for the information at runtime you must set the Default value of each parameter to the value you want to use.

1) ***Click*** the **Startup Info** tab on the **Experiment Object Properties** dialog.

2) ***Click*** the **checkbox** associated with each of the **Age**, **Sex**, **Handedness**, and **ResearcherID** parameters to include these parameters in the experiment.

3) ***Click*** the ? icon beside each of the same parameters to enable prompting for the parameter at runtime.

4) ***Click*** the **OK** button to accept the changes and dismiss the dialog.



Task 5: Add a Net Station package call to initialize the system

Add a package call at the beginning of the SessionProc to initialize the system for use with Net Station. Name the object "NSInit".

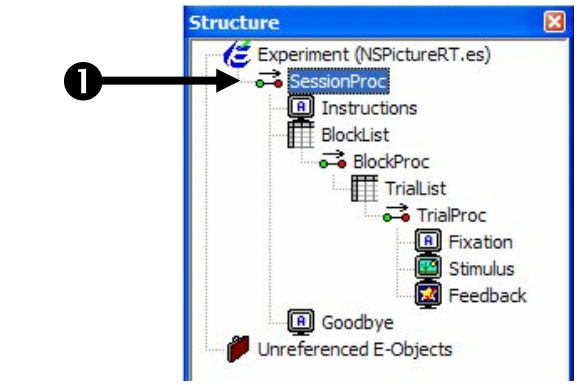
The Net Station package file must be initialized at the start of the experiment by making a call to the NetStation_Init routine. To call a routine in a package file you can either make the call directly using E-Basic script within an Inline object, or (more preferably) drag a PackageCall object from the E-Studio Toolbox, drop it at the desired location within the experiment, and edit its properties. When using the PackageCall object it is strongly recommended that you rename the object to reflect the specific package file and routine that are being referenced within the object. When creating paradigms for use with Net Station it is a common convention to create a name for each PackageCall object by abbreviating Net Station as "NS" and then concatenating the name of the routine being called, e.g. the object which is used in this task to call the "NetStation_Init" routine is named "NSInit".

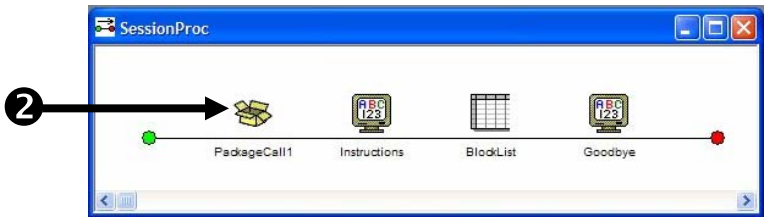
1) ***Double click*** the **SessionProc** object to open it in the workspace.

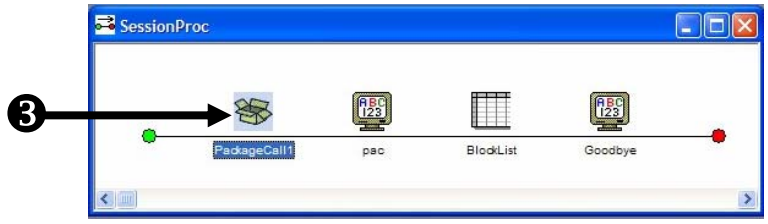
2) ***Drag*** a new **PackageCall** object from the **Toolbox** and ***drop*** it as the first object in the **SessionProc** procedure. The object will be given a default name of **PackageCall1**.

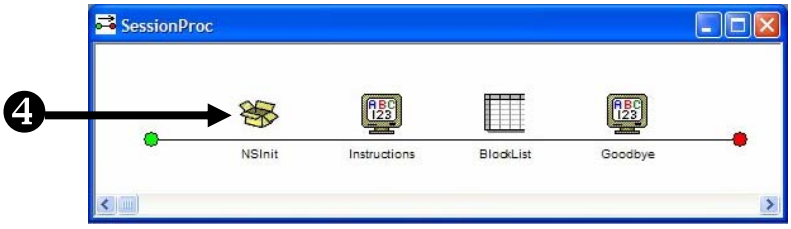
3) ***Click*** on the **PackageCall1** object to select it then ***press* F2** to rename the object. *You may alternatively right click on the object and select Rename from the context menu.*

4) ***Type* "NSInit"** as the new object name and then ***press* Enter** to accept the change.









Task 5 (continued): Add a Net Station package call to initialize the system

Configure the NSInit PackageCall object to call the Init routine of the Net Station package file and accept the default Parameters list.

You use the Properties dialog of the PackageCall object to specify which Package File and Routine are to be called in each instance. After you select a Package within the interface, the Routine dropdown list will be populated with all of the routines contained within the package that you can call via a PackageCall object. After you select a Routine, the Parameters field will be set to the default parameters and the Description field will be filled with the text that the package file author included for the selected routine. Unless you are using the default communications parameters found in the NETSTATION.INI file, you must edit the parameters to specify socket support and the IP address that you are currently using. You can refer to the Description field for information about each parameter in the list (any parameter in double quotes indicates string data).

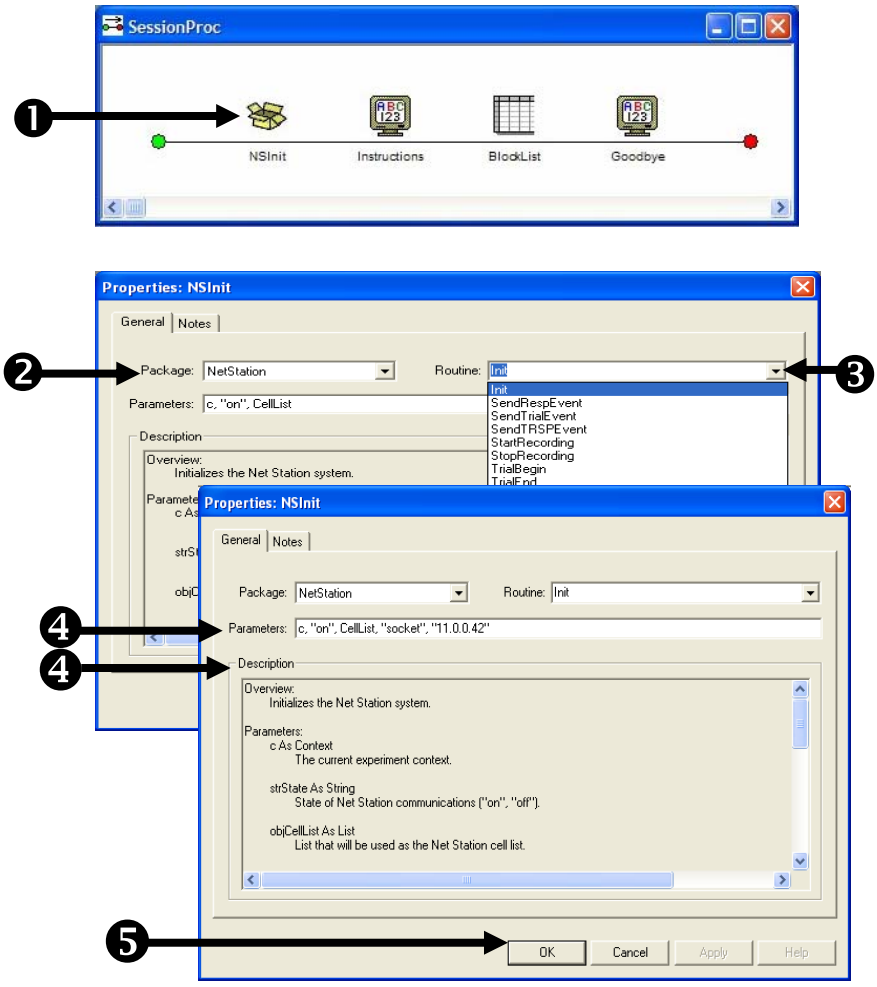
1) ***Double click*** click the **NSInit** PackageCall object within the **SessionProc** to display its **Properties** dialog.

2) ***Select*** **NetStation** from the **Package** dropdown list.

3) ***Select*** **Init** from the **Routine** dropdown list.

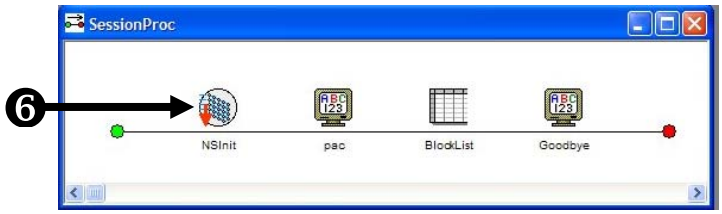
4) ***Review*** the parameters listed in the **Parameters** and **Description** fields. ***Update*** the **Parameters** to specify **"socket"** communications and correct IP address.
The IP address shown is an illustration and should be replaced with the IP address of your Net Station computer.

5) ***Click*** the **OK** button to accept the changes and dismiss the dialog.



Task 5 (continued): Add a Net Station package call to initialize the system

6) If the Package file defines an icon for the routine it will replace the default PackageCall icon, otherwise the icon associated with the entire package file itself will be used.



Task 6: Add a Net Station package call to uninitialize the system

Add a package call at the end of the SessionProc to uninitialize the system by calling the Uninit routine within the Net Station package file. Rename the object "NSUninit".

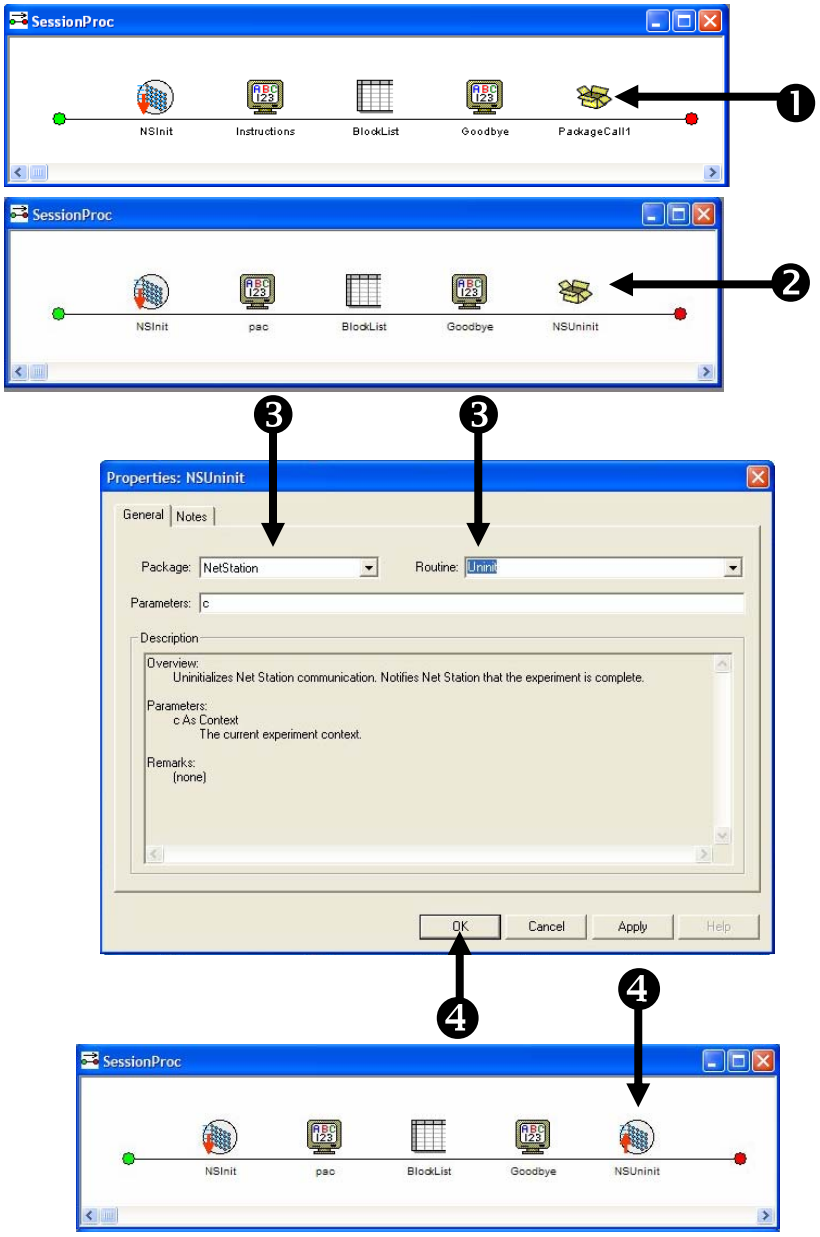
At the end of a Net Station enabled experiment the system must be uninitialized. The NetStation_Uninit call will send a message to Net Station to indicate that the experiment is now complete. This call is typically placed at the end of your SessionProc. After you have placed and configured the object notice that the icons for the NSInit and NSUninit objects are a matched pair, e.g. they each show a small arrow pointing in opposite directions which is intended to indicate the start and end of an experiment level.

1) ***Drag*** a new **PackageCall** object from the **Toolbox** and ***drop*** it as the last object in the **SessionProc** procedure.

2) ***Click*** on the **PackageCall1** object to select it then ***press* F2** to rename the object. ***Type* "NSUninit"** as the new object name and then ***press* Enter** to accept the change.

3) ***Double click*** the **NSUninit** object to open its **Properties** dialog. Set the **Package** field to **NetStation** and the **Routine** field to **Uninit**.

4) ***Click*** the **OK** button to accept the changes and ***verify*** the object has been added to the end of the **SessionProc**. *Note that the icons for the NSInit and NSUninit objects are a matched pair.*



Task 7: Add a Net Station package call to start recording

Add a package call at the beginning of the BlockProc to direct Net Station to start recording. Name the object "NSStartRecording".

When you want Net Station to begin recording you must direct it to do so by making a call to the NetStation_StartRecording routine. You may start and stop recording as many times as you would like within your experiment. Typically you will start recording at the beginning of a block of trials and stop recording at the end of the block. Open the BlockProc Procedure and add a new PackageCall object at the beginning of the Procedure (i.e. immediately before the TrialList object). Rename this object as "NSStartRecording" and set its properties so that it calls the NetStation_StartRecording routine.

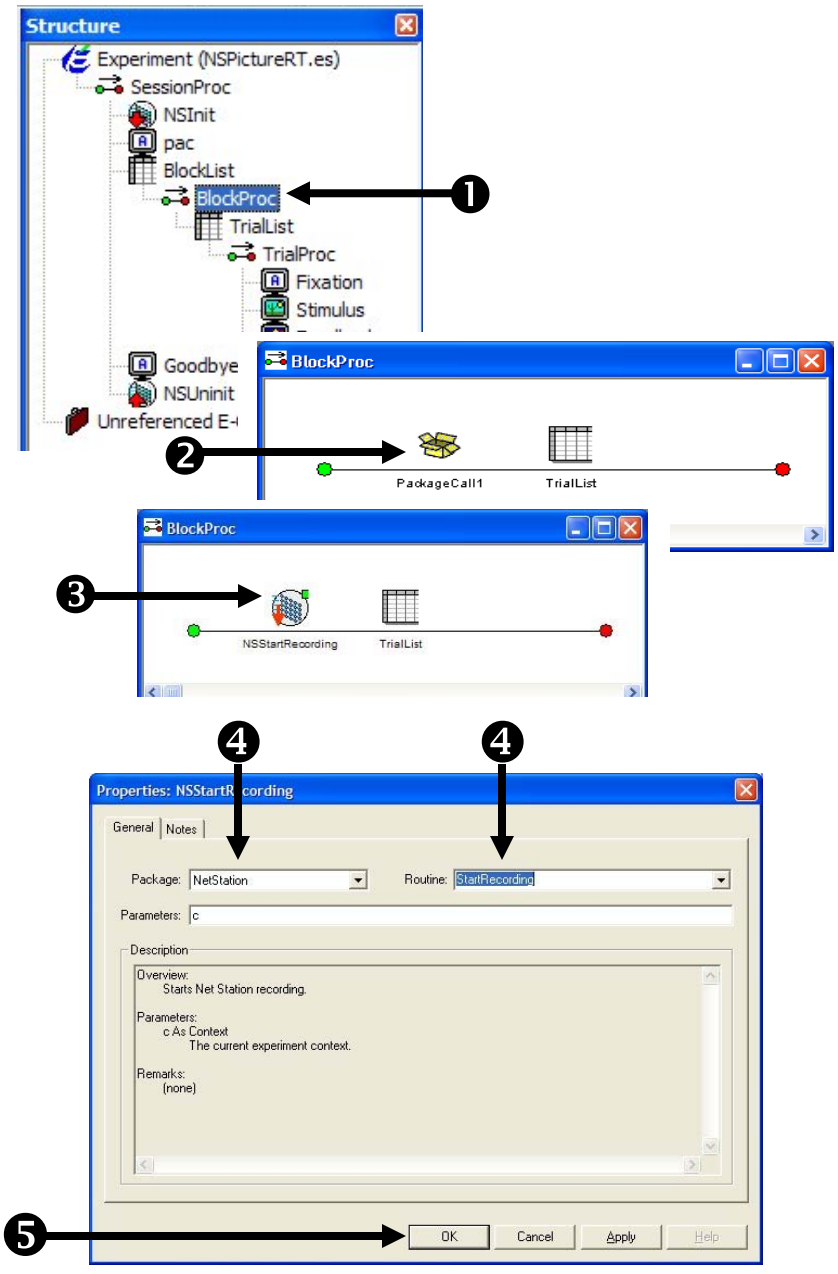
1) ***Double click*** the **BlockProc** object to open it in the workspace.

2) ***Drag*** a new **PackageCall** object from the **Toolbox** and ***drop*** it as the first object in the **BlockProc** procedure.

3) ***Click*** on the new object to select it then ***press*** **F2** and ***rename*** the object as **"NSStartRecording"**.

4) ***Double click*** on the **NSStartRecording** object to open its dialog then set the **Package** field to **NetStation** and the **Routine** field to **StartRecording**.

5) ***Click*** the **OK** button to accept the changes and dismiss the dialog.
Note, due to the processing time required to start and stop recording within Net Station, starting and stopping recording for each trial is not recommended.



Task 8: Add a Net Station package call to stop recording

Add a package call at the beginning of the BlockProc to direct Net Station to start recording. Name the object "NSStartRecording".

When you want Net Station to stop recording you must direct it to do so by making a call to the NetStation_StopRecording routine. Add a new PackageCall object at the end of the BlockProc Procedure (i.e. immediately after the TrialList object). Rename this object as "NSStopRecording" and set its properties so that it calls the NetStation_StopRecording routine.

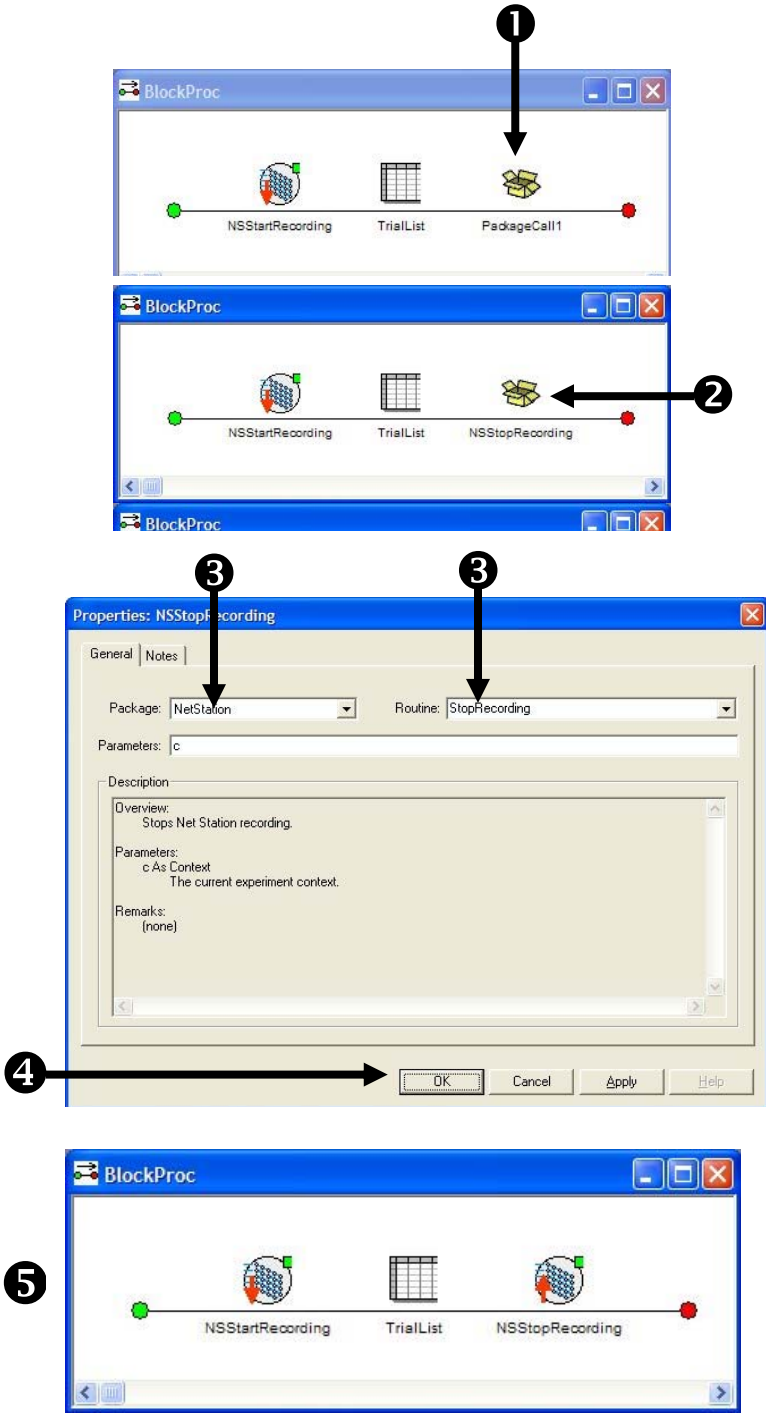
1) ***Drag*** a new **PackageCall** object from the **Toolbox** and ***drop*** it as the last object in the **BlockProc** procedure.

2) ***Click*** on the new object to select it then ***press*** **F2** and ***rename*** the object as **"NSStopRecording"**.

3) ***Double click*** on the **NSStopRecording** object to open its **Properties** dialog then set the **Package** field to **NetStation** and the **Routine** field to **StopRecording**.

4) ***Click*** the **OK** button to accept the changes and dismiss the dialog.

5) ***Verify*** that your **BlockProc** Procedure matches the picture.



Task 9: Add a Net Station package call to initialize the trial

Add a package call at the beginning of the TrialProc to initialize each trial. Name the object "NSTrialBegin".

Net Station must be notified of the start of each trial. Open the TrialProc Procedure and add a new PackageCall object at the beginning of the Procedure (i.e. immediately before the Fixation object). Rename this object as "NSTrialBegin" and set its properties so that it calls the NetStation_TrialBegin routine.

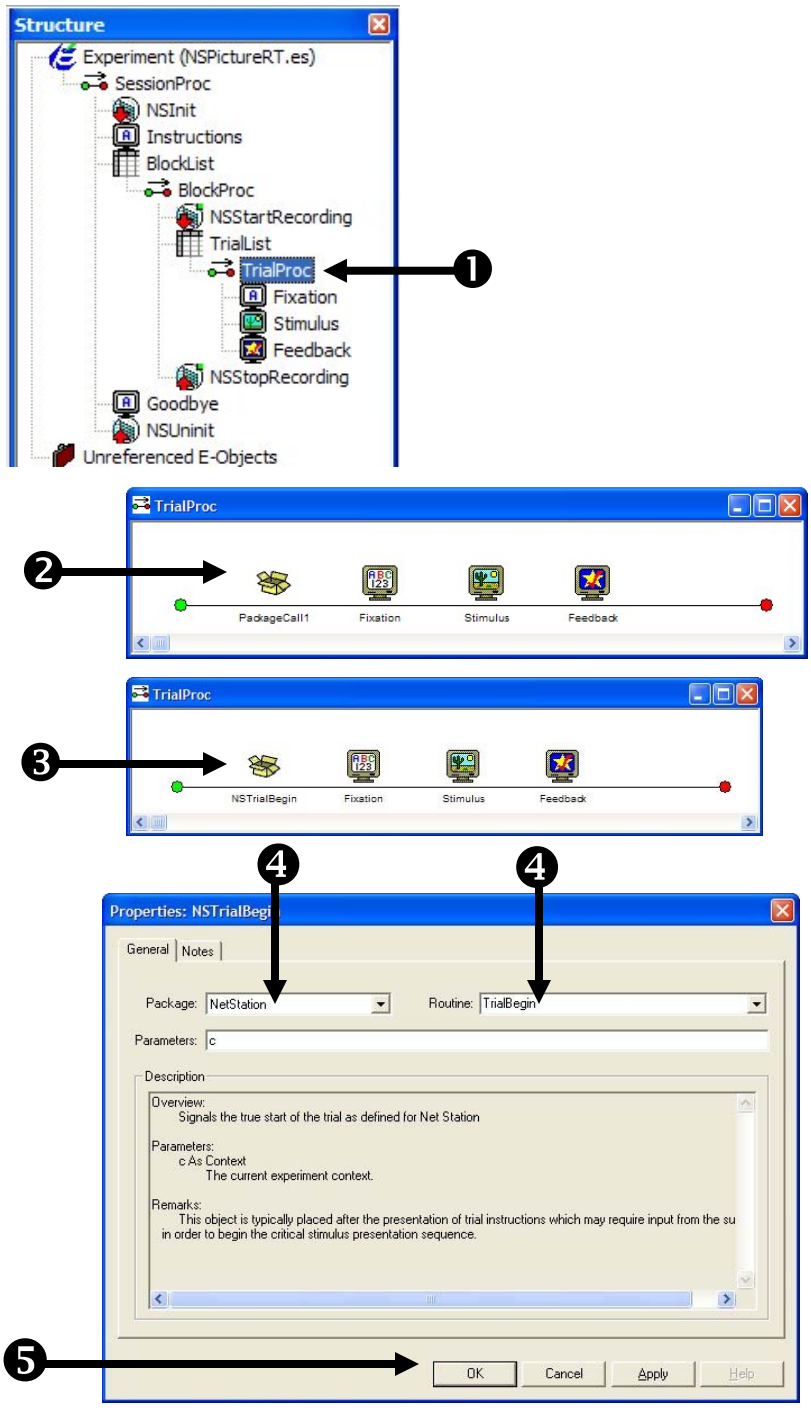
1) ***Double click*** the **TrialProc** object to open it in the workspace.

2) ***Drag*** a new **PackageCall** object from the **Toolbox** and ***drop*** it as the first object in the **TrialProc** procedure.

3) ***Click*** on the new object to select it then ***press* F2** and ***rename*** the object as **"NSTrialBegin"**.

4) ***Double click*** on the **NSTrialBegin** object to open its **Properties** dialog then set the **Package** field to **NetStation** and the **Routine** field to **TrialBegin**.

5) ***Click*** the **OK** button to accept the changes and dismiss the dialog.



Task 10: Add a Net Station package call to uninitialize the trial

Add a package call at the end of the TrialProc to uninitialize each trial. Name the object "NSTrialEnd".

Net Station must be notified of the end of each trial. Add a new PackageCall object at the end of the Procedure (i.e. immediately after the Feedback object). Rename this object as "NSTrialEnd" and set its properties so that it calls the NetStation_TrialEnd routine.

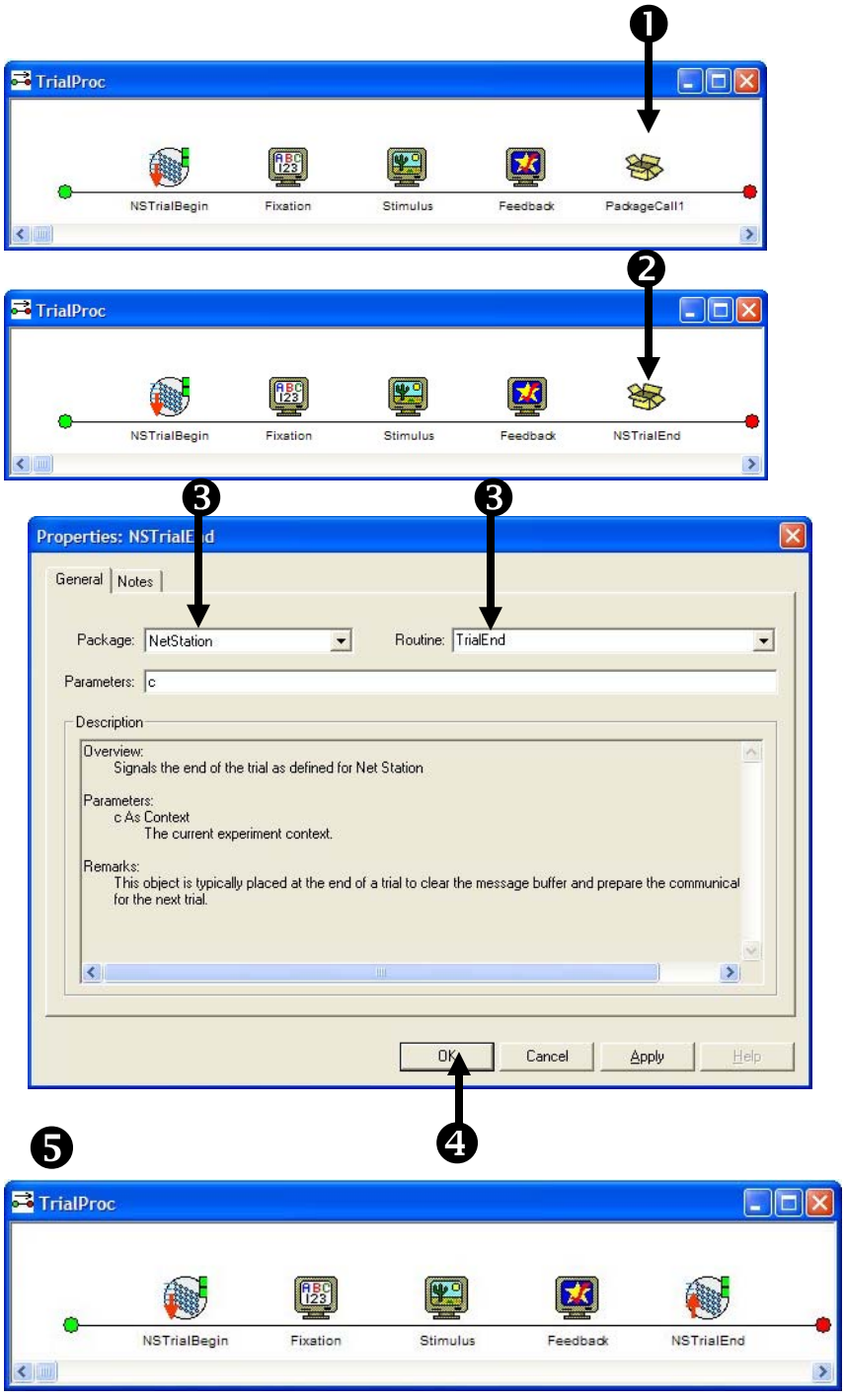
1) ***Drag*** a new **PackageCall** object from the **Toolbox** and ***drop*** it as the last object in the **TrialProc** procedure.

2) ***Click*** on the new object to select it then ***press*** **F2** and ***rename*** the object as **"NSTrialEnd"**.

3) ***Double click*** on the **NSTrialEnd** object to open its **Properties** dialog then set the **Package** field to **NetStation** and the **Routine** field to **TrialEnd.**

4) ***Click*** the **OK** button to accept the changes and dismiss the dialog.

5) ***Verify*** that your **TrialProc** Procedure matches the picture.



Task 11: Add script to send trial events to Net Station

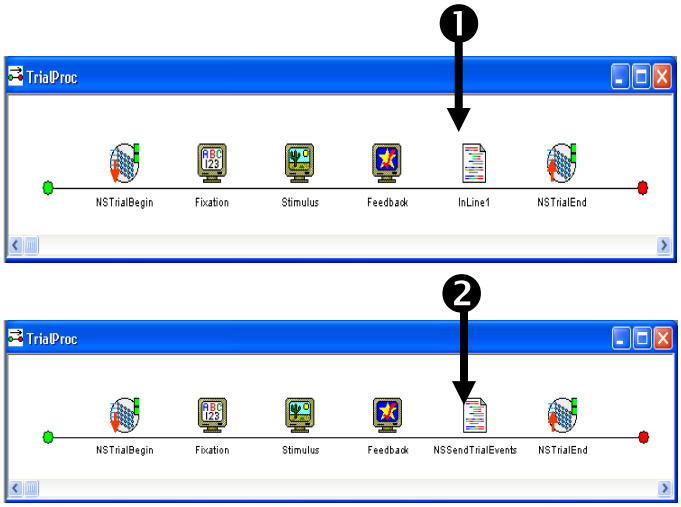
Add an Inline object after the Feedback object. Name the object "NSSendTrialEvents". Add script to the Inline to make direct package calls to send the required trial event information to Net Station.

Net Station requires that it be notified of all critical trial events (e.g. stimulus events presented or participant responses that were accepted during the trial). While you may use a series of PackageCall objects to complete this task, it is often easier to just use a single Inline object and a few lines of E-Basic script to make a series of direct calls to the required package routines. Add a new Inline object after the Feedback object on the TrialProc. Rename this object as "NSSendTrialEvents" and open the object in the workspace so that you can type in the required script.

Net Station requires that it be notified of any participant response that was collected during each trial. You can do this by making a direct call to the NetStation_SendRespEvent routine. This routine requires that you pass in a reference to the object within the trial sequence that collected the response (e.g. the Stimulus object in this example). It is important to make sure that the allowable response interval within each trial has completed before making this call (e.g. if you are using PreRelease on objects within your trial sequence and a participant responds near the very end of the response interval). You could potentially send invalid or incomplete response data to Net Station (see *E-Prime User's Guide, Chapter 3: Critical Timing, Section 3.3.1.1* for details on how to avoid this problem.)

1) ***Drag*** an **InLine** object from the **Toolbox** and ***drop*** it after the **Feedback** object in the **TrialProc** procedure.

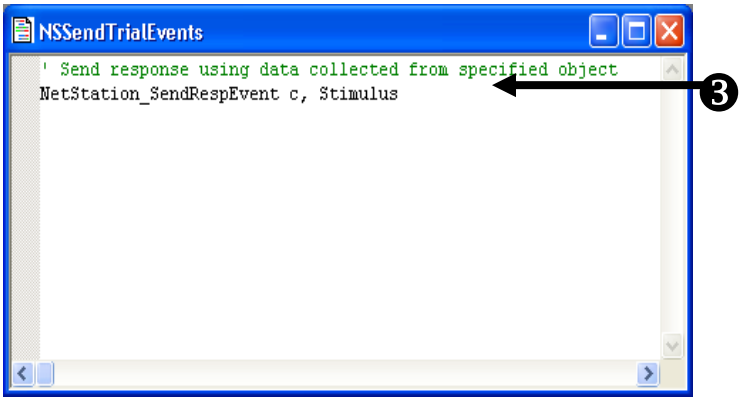
2) ***Click*** on the new object to select it then ***press*** **F2** and ***rename*** the object as **"NSSendTrialEvents"**.



Task 11 (continued): Add script to send trial events to Net Station

Add script to the Inline to make direct package calls to send the required trial event information to Net Station.

3) ***Double click*** on the **NSSendTrialEvents** object to open it in the workspace and then ***type*** in the **script** exactly as shown to add a direct call to the **NetStation_SendResp Event** routine.
It is recommended that you always add a comment line to document each direct package file call.



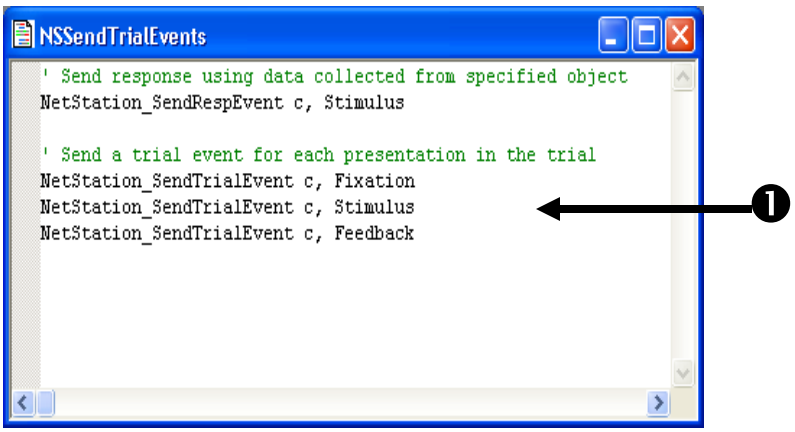
Task 11 (continued): Add script to send trial events to Net Station

Add script to the NSSendTrialEvents Inline to make direct package calls to send the required trial event information to Net Station.

Prior to the end of each trial, Net Station requires that it be sent all timing information related to any critical stimulus presentation that occured within the trial. Append additional script to the NSSendTrialEvents Inline object to make a direct call to the NetStation_SendTrialEvent routine for each object of interest in the trial sequence of your experiment (e.g. in this experiment we are interested in the Fixation, Stimulus, and Feedback objects because they directly alter the visual display).

Note the syntax of the NetStation_SendTrialEvent subroutine call. Its required parameters are the current experiment Context (c) and a reference to the object that generated the current event. The routine also includes two additional optional parameters (KeyValue and KeyName) that are not explicitly used in this example. The first optional parameter, KeyValue, specifies a user defined value that should be associated with the current trial event. If the value is not specified it will default to 0. If the value is specified it must be numeric. The second optional parameter, KeyName, is a user defined four character code to associate with the KeyValue. If the KeyName is not specified it defaults to “argu”. If a KeyName is specified it must be exactly 4 characters in length.

1) ***Type*** in the **script** exactly as shown to add a call to **NetStation_SendTrial Event** routine for each stimulus presentation object in the trial.


NSSendTrialEvents
' Send response using data collected from specified object
NetStation_SendRespEvent c, Stimulus
' Send a trial event for each presentation in the trial
NetStation_SendTrialEvent c, Fixation
NetStation_SendTrialEvent c, Stimulus
NetStation_SendTrialEvent c, Feedback
1

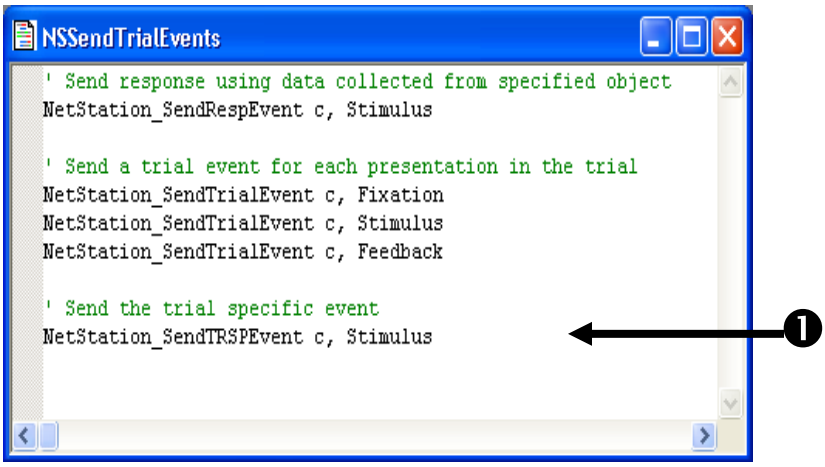
Task 11 (continued): Add script to send trial events to Net Station

Add script to the NSSendTrialEvents Inline to send the TRial SPecific (TRSP) event to Net Station.

Net Station requires that a TRial SPecific (TRSP) event be sent prior to the end of each trial so that it can be notified of behavioral data related to the critical stimulus and response. Append an additional line of script to the NSSendTrialEvents Inline object to make a direct call to the NetStation_SendTRSPEvent routine.

Note the syntax of the NetStation_SendTRSPEvent subroutine call. Its required parameters are the current experiment Context (c) and a reference to the object that collected the participant's response (e.g. the Stimulus object in this example). The routine also includes an additional optional parameter (KeyList) that is not currently used in this example. The optional KeyList parameter may be used to specify a reference to a user defined List object within the experiment that has been configured to describe additional key data that should be associated with the current trial. If the KeyList parameter is not specified then no additional key data will be sent to Net Station as part of the TRSP event. Adding user defined keys using a KeyList will be covered later in Tutorial 2.

1) ***Type*** in the **script** exactly as shown to add a call to the **NetStation_SendTRSP Event** routine.


NSSendTrialEvents
' Send response using data collected from specified object
NetStation_SendRespEvent c, Stimulus
' Send a trial event for each presentation in the trial
NetStation_SendTrialEvent c, Fixation
NetStation_SendTrialEvent c, Stimulus
NetStation_SendTrialEvent c, Feedback
' Send the trial specific event
NetStation_SendTRSPEvent c, Stimulus
1

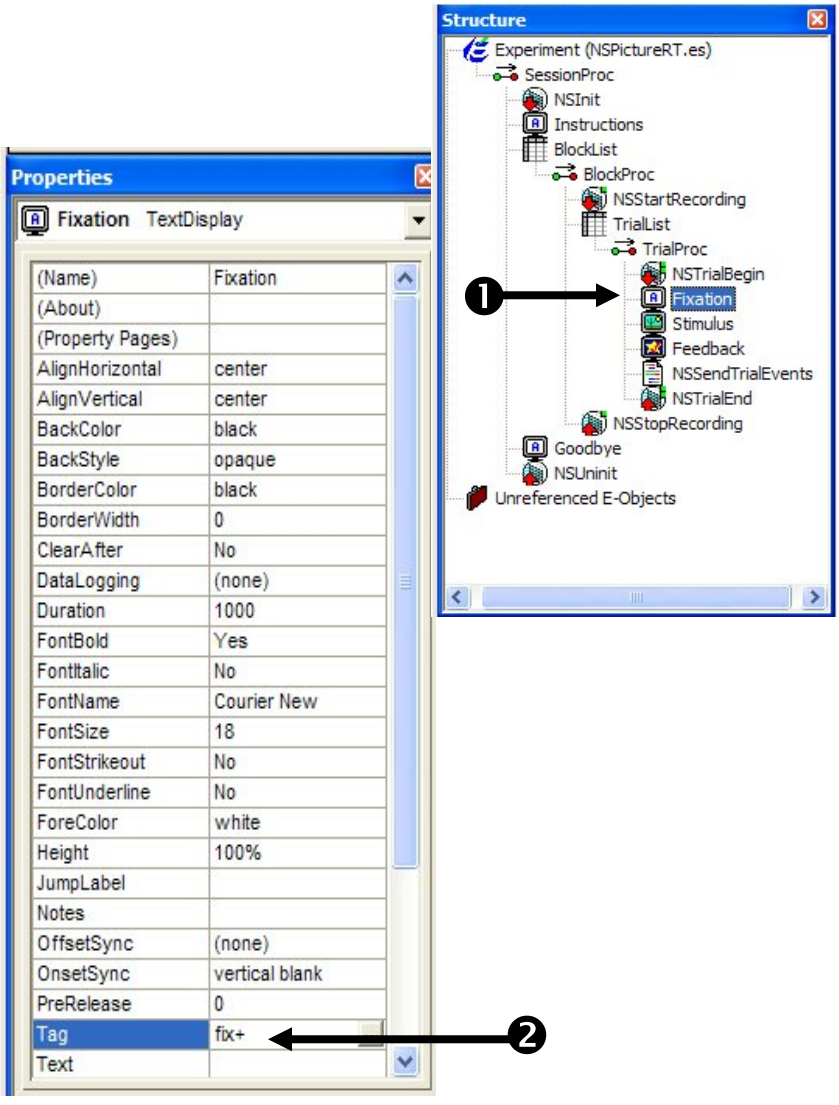
Task 12: Assign a unique four character code to each object of interest

Assign a unique four character code to each stimulus presentation object or event of interest within the trial sequence. Set the Tag property of the Fixation object to "fix+".

Net Station requires that each event of interest within a trial sequence be identified via a unique four character code. Within a Net Station enabled experiment you assign these unique codes to objects by setting each object's "Tag" property. At runtime, routines within the Net Station Package File will examine the assigned Tag property of an object in order to associate data sent to Net Station with a particular object. In this example experiment you need to set the Tag property of the Fixation, Stimulus, and Feedback objects to "fix+","stm+","stm-" respectively. Note that the editing of an object's Tag property can only be done through the Properties window within E-Studio so you should verify that window is currently viewable before beginning this task.

1) ***Click*** on the **Fixation** object in the **Structure** view to select it and display its properties in the **Properties window.**
Note, to view the Properties window choose View>Properties from the application menu bar.

2) ***Click*** in the **edit box** beside the **Tag** property and ***type*** "**fix+**" then ***press*** the **Enter** key to accept the change.
Note that assigned codes must be exactly 4 characters in length or an error will be generated when the experiment is run with Net Station.



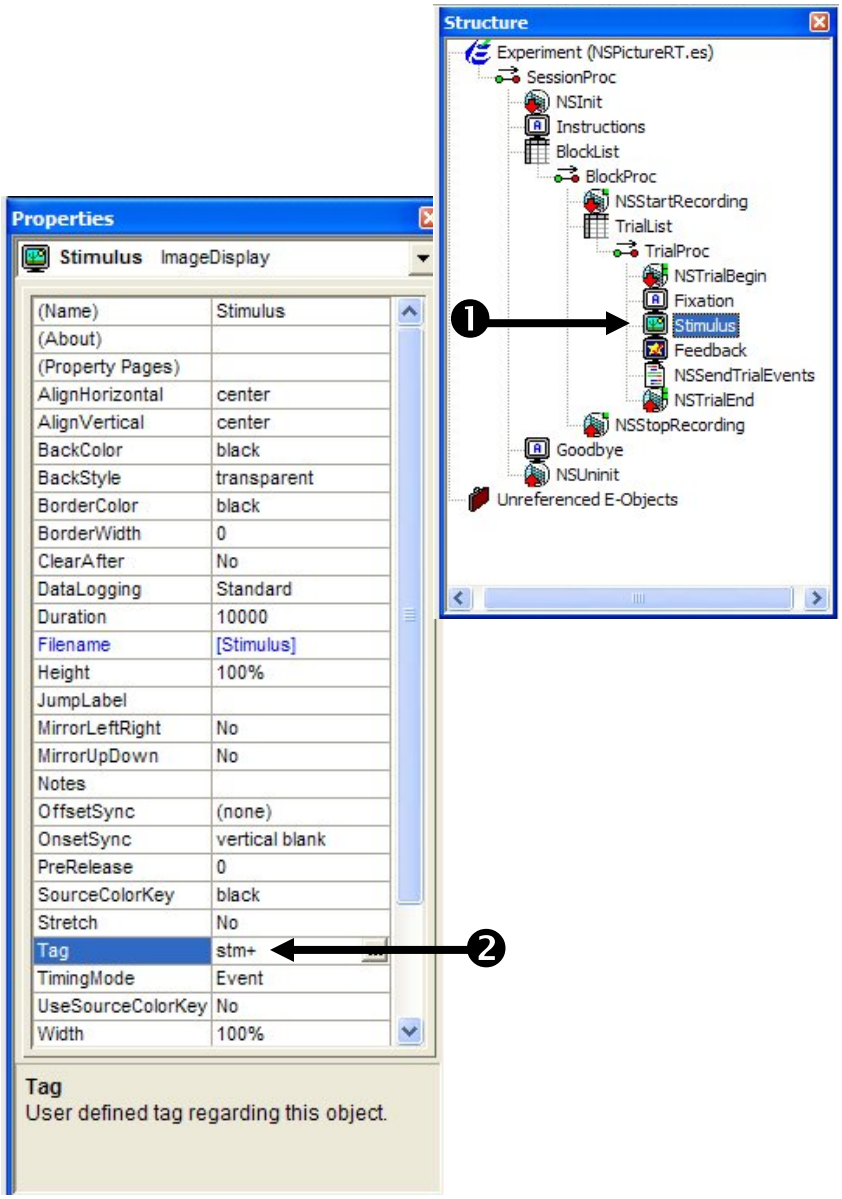
Task 12 (continued): Assign a unique four character code to each object of interest

Set the Tag property of the Stimulus object to "stm+".

In this example the Stimulus object is responsible for presenting the critical stimulus and also for enabling and accepting a response from the participant. Select the Stimulus object and set its Tag property to the four character code of "stm+" to indicate that it will be the object that presents the critical stimulus.

1) ***Click*** on the **Stimulus** object in the **Structure** view to select it and display its properties in the **Properties Window.**

2) ***Click*** in the **edit box** beside the **Tag** property and ***type*** "**stm+**" then ***press*** the **Enter** key to accept the change.
Note that it is a common convention in Net Station enabled experiments to append a "+" to the end of an object's assigned tag to indicate the start/onset of an event.



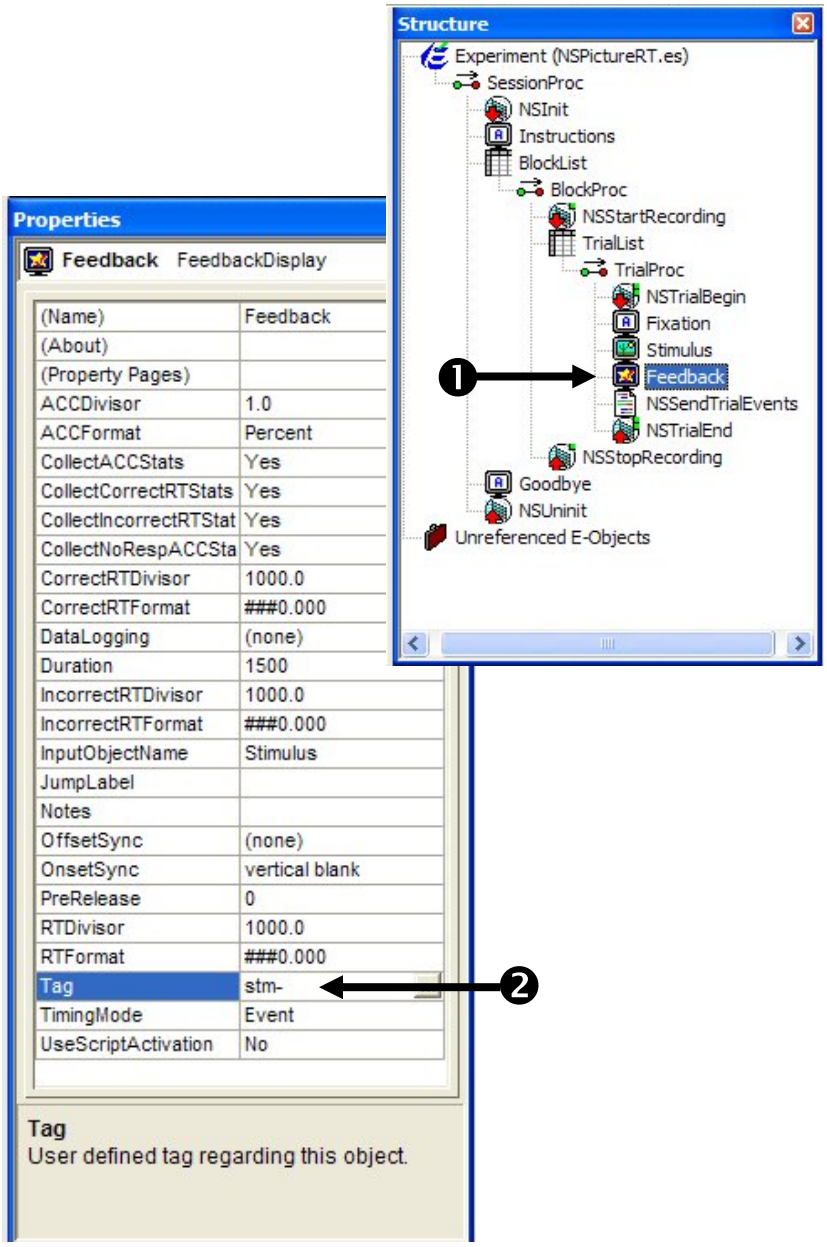
Task 12 (continued): Assign a unique four character code to each object of interest

Set the Tag property of the Feedback object to "stm-".

In this example the Feedback object is configured to run after the participant has responded (or alternatively after the Stimulus object has timed out). When the Feedback object runs it clears the screen and thus removes/terminates the display of the critical stimulus. Select the Feedback object and set its Tag property to the four character code of "stm-" to indicate that it is the object that signals the end of the critical stimulus.

1) ***Click*** on the **Feedback** object in the **Structure** view to select it and display its properties in the **Properties Window.**

2) ***Click*** in the **edit box** beside the **Tag** property and ***type*** "**stm-**" then ***press*** the **Enter** key to accept the change.
Note that it is a common convention in Net Station enabled experiments to append a "-" to the end of an object's assigned tag to indicate the end/offset of an event.



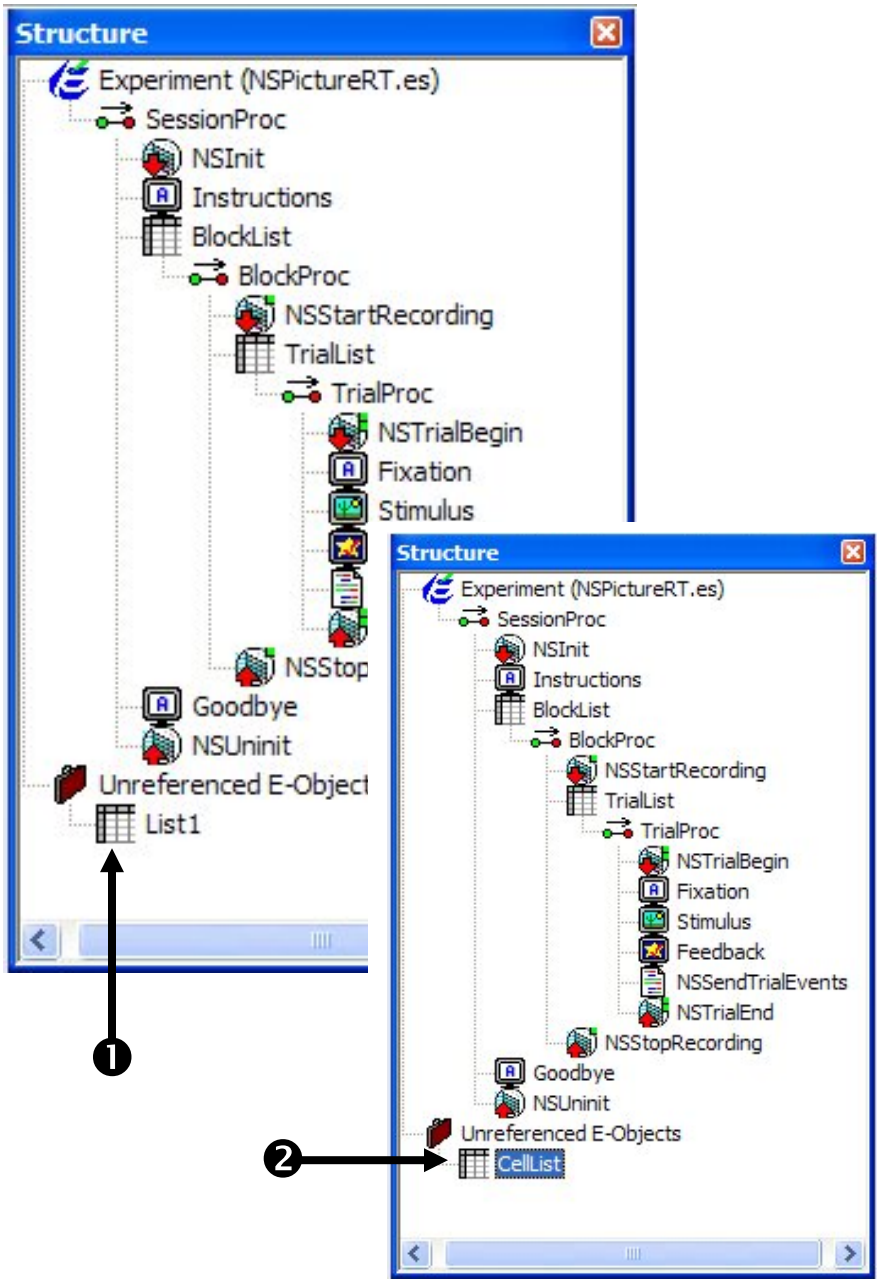
Task 13: Add a CellList object

Add a new List object to the Unreferenced E-Objects branch of the tree. Name the object "CellList".

Net Station identifies conditions or "cells" within an experiment by referencing user assigned Cell Numbers and Cell Labels. The routines within the Net Station Package File lookup information for these cells by examining the attributes of a named Cell List contained within the experiment. A Cell List is just a List object that has been created in the experiment and configured with a specified set of named attributes. Create a new List object in the Unreferenced E-Objects branch of the Structure view and rename it as "CellList".

1) ***Drag*** a new **List** object from the **Toolbox** and ***drop*** it on the **Unreferenced E-Object** branch of the **Structure** view.

2) ***Click*** on the new **List1** object to select it then ***press* F2** and ***rename*** the object as **"CellList"**.



Task 14: Add the required CellNumber and CellLabel attributes to the CellList.

Open the CellLIst object in the workspace and add two new attributes and one new level.

The CellList must minimally contain two named attributes - “CellNumber” and “CellLabel”. Open the CellList object in the workspace and add two new attributes. Since there are two cells defined and used in this experiment you must also add an additional level/row to the CellList.

1) ***Double click*** the **CellList** object to open it in the workspace.

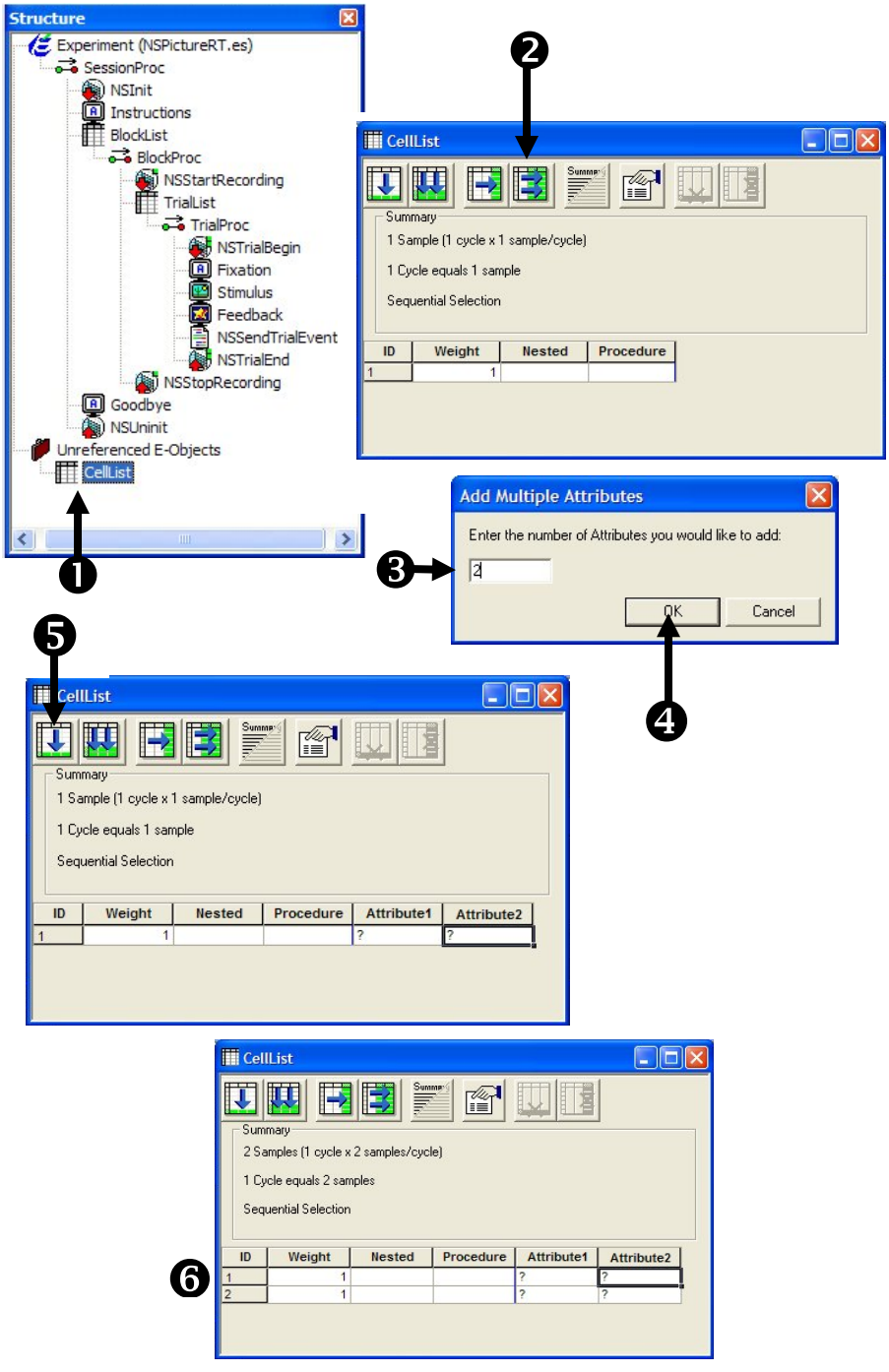
2) ***Click*** the **Add Multiple Attributes** button.

3) ***Type*** **2** in the **Add Multiple Attributes** dialog.

4) ***Click*** the **OK** button to add two new attributes to the **CellList**.

5) ***Click*** the **Add Level** button to add an additional level/row to the **CellList**.

6) ***Verify*** that your **CellList** matches the picture.



Task 14 (continued): Add the required CellNumber and CellLabel attributes to the CellList

Rename the new attributes to CellNumber and CellLabel then edit the attribute values to define the cells used within the experiment.

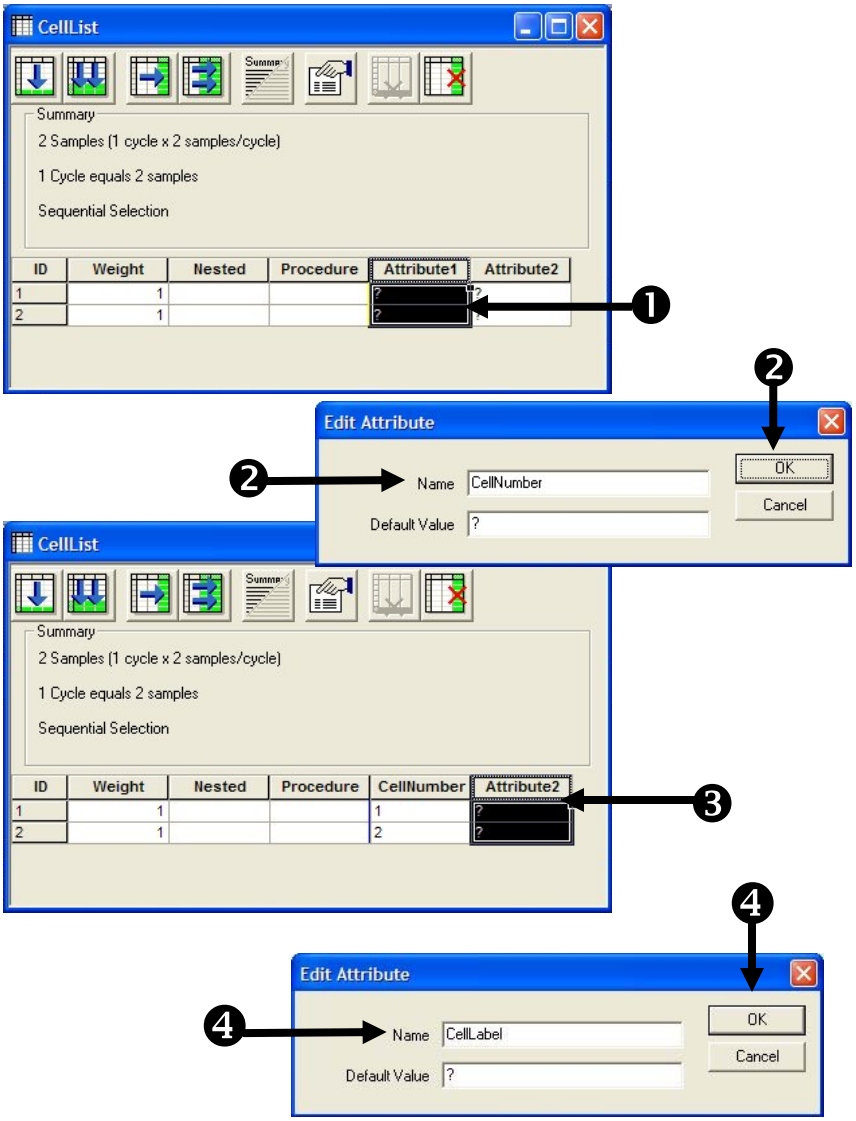
The routines within the Net Station Package File expect the CellList to contain attributes named "CellNumber" and "CellLabel". Rename each attribute respectively. If you misspell or forget to include one of these attributes, the Net Station Package File will generate an error when you attempt to run the experiment. Values for the CellNumber attribute must be numeric and must be unique within the experiment. You may currently define up to 255 unique cells within an experiment. Values for the CellLabel can be any string you choose to identify and describe the cell (i.e. CellLabels do not have to be four character codes).

1) ***Double click*** the header of the **Attribute1** column to display the **Edit Attribute** dialog.

2) ***Type*** "**CellNumber**" in the **Name** field of the **Edit Attribute** dialog and ***click*** the **OK** button to accept the change.

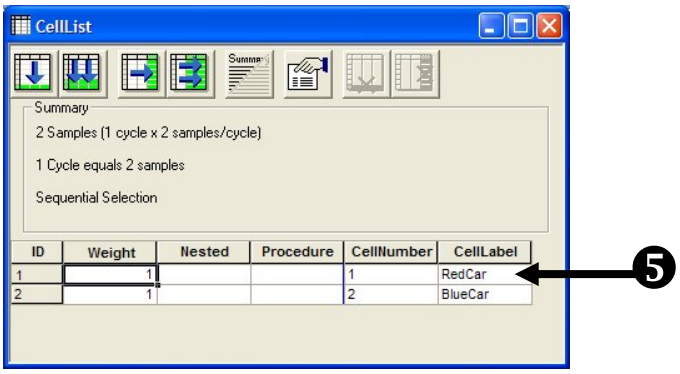
3) ***Double click*** the header of the **Attribute2** column to display the **Edit Attribute** dialog.

4) ***Type*** "**CellLabel**" in the **Name** field of the **Edit Attribute** dialog and ***click*** the **OK** button to accept the change.



Task 14 (continued): Add the required CellNumber and CellLabel attributes to the CellList

5) ***Click*** in each cell under the **CellNumber** and **CellLabel** attributes and ***edit*** the values to match those shown.
Note, it is recommended that you assign CellNumbers sequentially in order to maintain compatibility with most prior releases of Net Station.



Task 15: Add the required CellNumber attribute to the TrialList

Add a CellNumber attribute to the TrialList. Assign values to the CellNumber attribute to identify the specific cells represented in the TrialList.

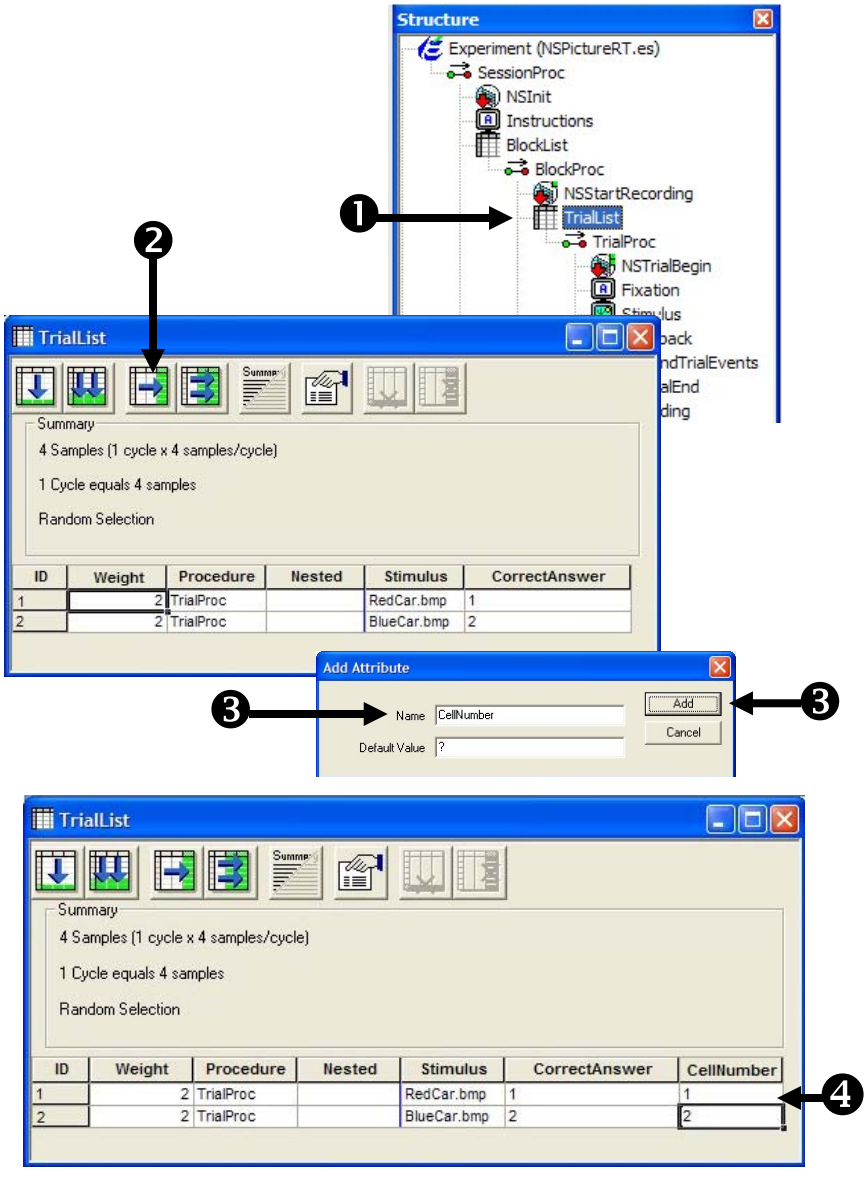
While you may define many different types of cells within an experiment, each individual trial must be associated with a single specific cell when it is run. You associate a trial with a particular cell by adding a CellNumber attribute to your trial level list (e.g. the TrialList object in this example) and then setting its values to identify the CellNumber of appropriate cell that is related to each trial represented in the list. Open the TrialList object in the workspace, create a new CellNumber attribute and assign its values. Note that there is no need to also create or include a CellLabel attribute on the trial level list (e.g. since the CellNumber is identified for each trial, the Net Station Package File can retrieve the cooresponding CellLabel as needed from the CellList that was defined earlier.)

1) ***Double click*** the **TrialList** object to open it in the workspace.

2) ***Click*** the **Add Attribute** button to add a new attribute to the **TrialList**.

3) ***Type*** "**CellNumber**" in the **Name** field of the **Add Attribute** dialog and ***click*** the **Add** button to create the new attribute.

4) ***Click*** in each cell under the **CellNumber** attribute and edit the values to match those shown in the picture.



Task 16: Run the experiment

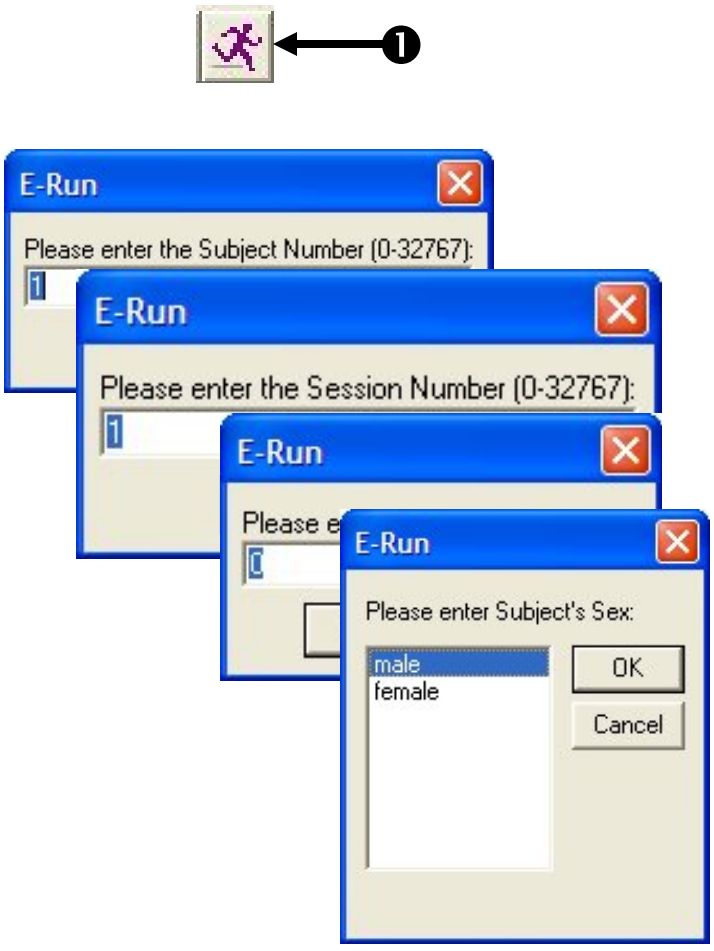
Run the experiment to verify that the appropriate prompts are presented and no runtime errors are generated.

Your experiment should now be enabled for use with Net Station! Net Station enabled experiments can be run locally from E-Studio during development and testing (i.e. without having the Net Station hardware connected). You should fully test your experiment locally before deploying it to the experiment presentation computer connected to your Net Station hardware. Note that while many Net Station related errors can be detected while testing locally, some errors will only be detected or generated when the actual Net Station hardware is in use. Thus, you should fully test your experiment with the Net Station hardware connected and active prior to scheduling actual participants.

Once you have launched the experiment you will be presented with a series of initialization prompts. Press Enter or click OK to accept the default values during testing. Read and respond to any instructions presented then observe the stimulus presentation sequence to verify the experiment is functioning correctly and that no errors are generated. If you have Net Station connected and active you should also use the Experiment Control Status Panel within the Workbench to review and verify the valid communications of Net Station related events.

1) ***Click*** the **Run icon** on the application toolbar to generate and run the experiment locally.

2) ***Press* Enter** to accept the default values for each of the initialization prompts presented.

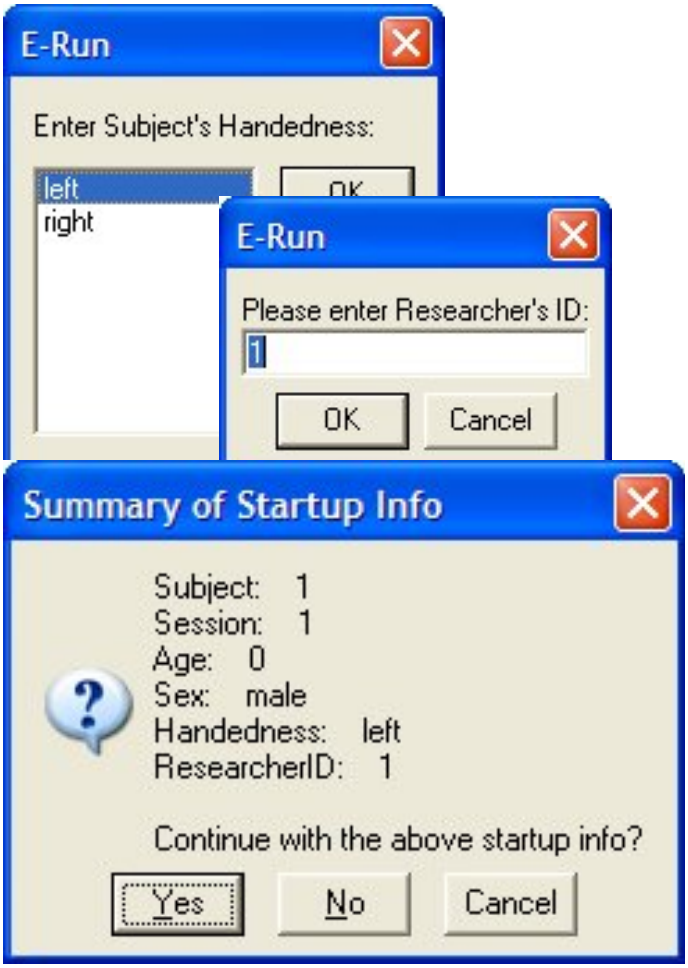


Task 16 (continued): Run the experiment

Run the experiment to verify that the appropriate prompts are presented and no runtime errors are generated.

3) ***Observe*** the stimulus presentation sequence to verify the experiment is functioning correctly and completes with no errors being generated. *Note that when testing without the Net Station hardware connected you will receive a message during initialization indicating that Net Station communications will be disabled. While this is the expected functionality when testing locally, if this message occurs with the Net Station hardware connected and active it indicates a communications problem that must be resolved.*

4) ***Save*** the experiment.



Tutorial 2 – Adding user defined keys

Summary: Net Station allows you to associate user defined keys and values with any particular trial through the use of the TRial SPecific (TRSP) event. To enable this feature you create and define a new "KeyList" within your experiment and add a reference to the KeyList in each call to the Net Station Package routine that is responsible for sending the TRSP event to Net Station. At runtime, the current value of all user defined keys that are defined in the KeyList will be retrieved from the experiment Context and sent to Net Station as part of the TRSP event.

During this tutorial you may want to turn on the Net Station Workbench and have the Experiment Control Status Panel open and configured to view its contents in "Long Form".

Goal: This tutorial illustrates how to add a user defined KeyList to the NSPictureRT experiment created in Tutorial 1. The KeyList will include one new key (named "stim") that will send to Net Station a reference to the name of the actual stimulus that was presented on each trial.

Overview of Tasks:

- Add a KeyList to the Unreferenced E-Objects branch of the Structure view
- Configure the KeyList object to add the required attributes and user defined keys
- Add a reference to the KeyList object in the NetStation_SendTRSPEvent call
- Verify the overall experiment structure and run the experiment

Recommended readings: It is required that you have read through and completed Tutorial 1 before beginning this tutorial.

Estimated Tutorial Time: 10-20 minutes.

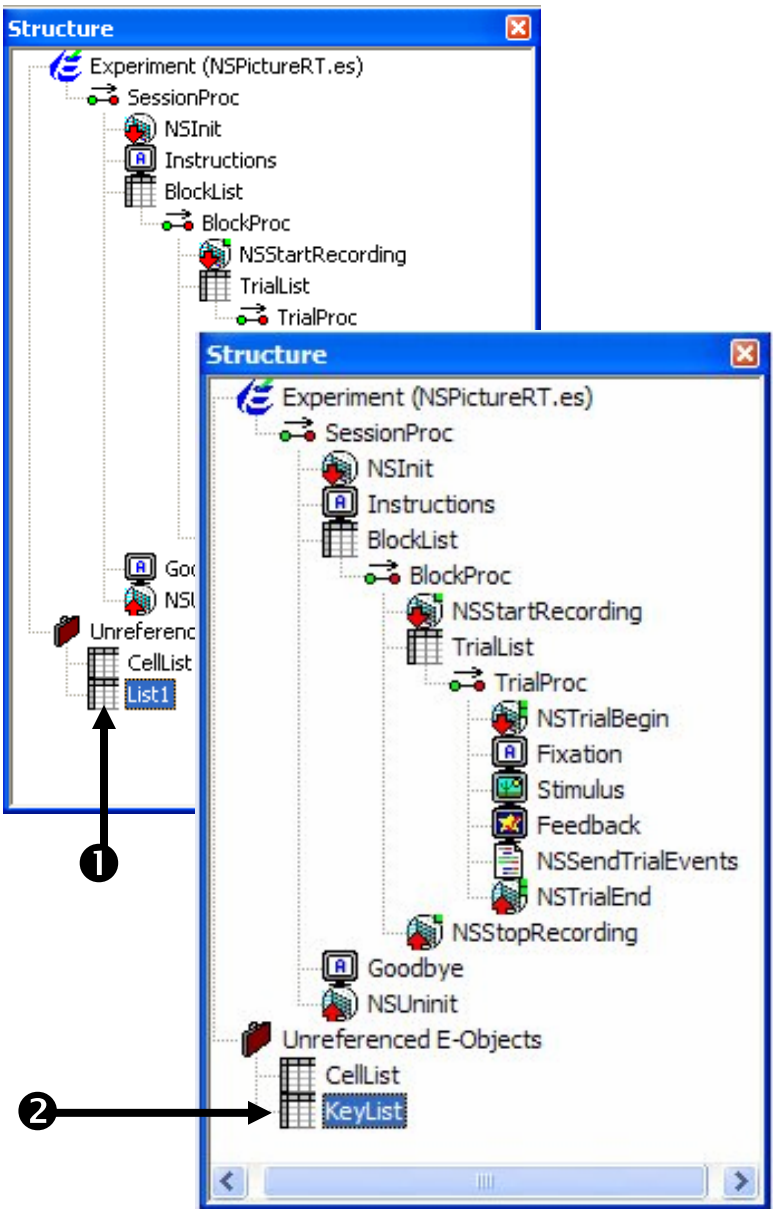
Task 1: Add a KeyList object

Add a new List object to the Unreferenced E-Objects branch of the Structure view. Name the object "KeyList".

A Net Station enabled experiment identifies user defined keys by the use of one or more "Key Lists" defined within the experiment. The routines within the Net Station Package File use these lists at runtime to lookup information (e.g. name, ID, data type, value) related to each user defined key. A KeyList is just a List object that has been created in the experiment and configured with a specified set of named attributes. Create a new List object in the Unreferenced E-Objects branch of the Structure view and rename it as "KeyList".

1) ***Open*** **NSPictureRT.es** and ***drag*** a new **List** object from the **Toolbox** and ***drop*** it on the **Unreferenced E-Objects** branch of the **Structure** view.

2) ***Click*** on the new **List1** object to select it then ***press*** **F2** to ***rename*** the object as **"KeyList"**.



Task 2: Add the required KeyName, KeyID, and KeyDataType attributes to the KeyList

Open the KeyList object in the workspace and add three new attributes.

Each Key List that is defined in an experiment must contain the three named attributes of “KeyName”, “KeyID”, “KeyDataType”. Open the KeyList object in the workspace and add three new attributes.

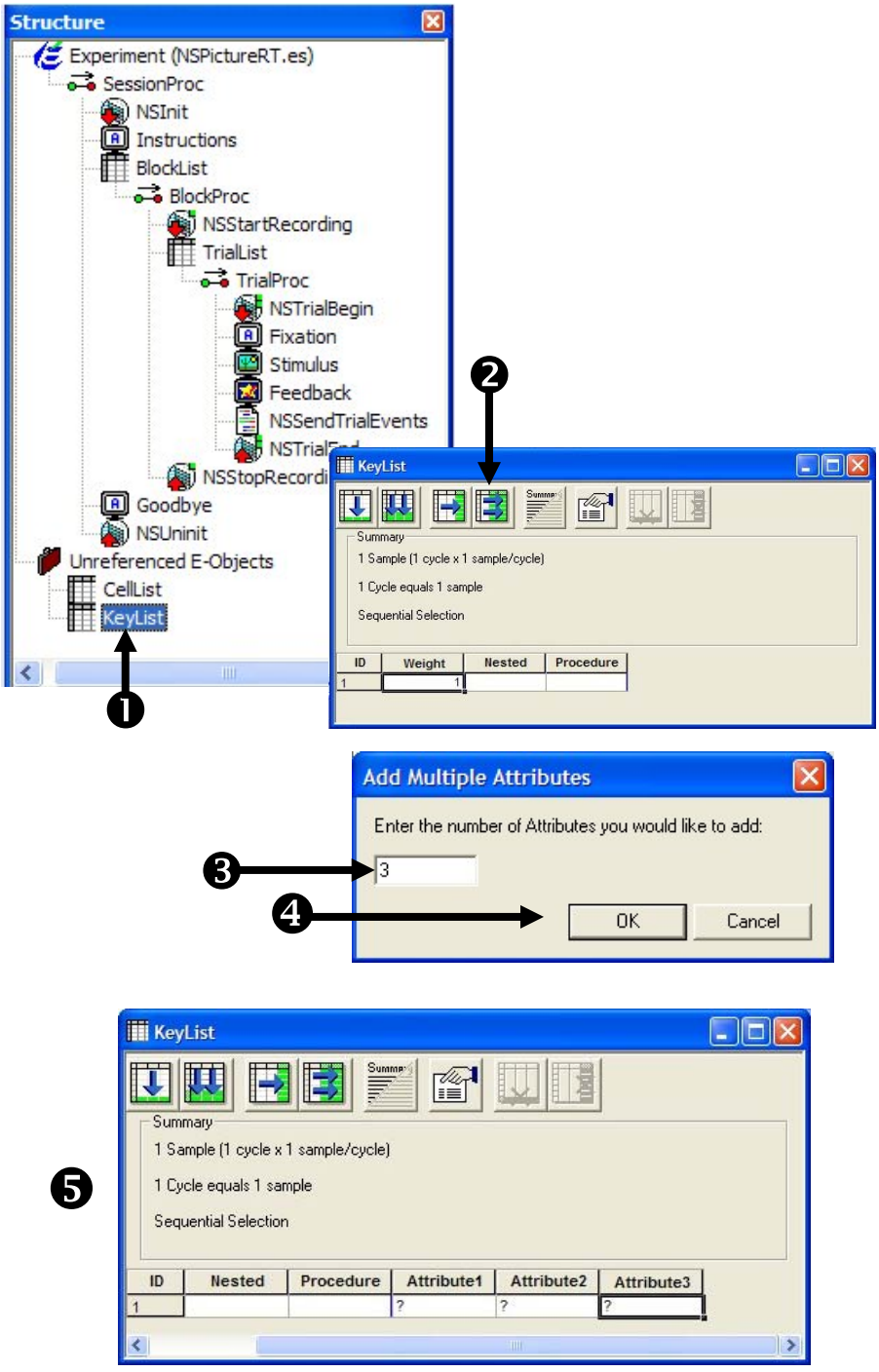
1) ***Double click*** the **KeyList** object to open it in the workspace.

2) ***Click*** the **Add Multiple Attributes** button.

3) ***Type*** **3** in the **Add Multiple Attributes** dialog.

4) ***Click*** the **OK** button to add three new attributes to the KeyList.

5) ***Verify*** that your **KeyList** matches the picture.



Task 2 (continued): Add the required KeyName, KeyID, and KeyDataType attributes to the KeyList

Rename the new attributes to KeyName, KeyID, and KeyDataType then edit the attribute values to create a new user defined key.

The routines within the Net Station Package File expect each List object that is used as a Key List to contain a set of attributes named “KeyName”, “KeyID”, and “KeyDataType”. Rename each attribute respectively. If you misspell or forget to include one of these attributes the Net Station Package File will generate an error when you attempt to run the experiment. The value of KeyName must be a string. The value of KeyID must be a four character code. The value of KeyDataType must be one of the following codes: “shor”, “long”, “bool”, “TEXT”, “sing”, “doub”, “type”. If the value of KeyDataType is left blank it will default to “shor”. Please refer to your Net Station documentation for more information on valid key values and ranges for the different data types.

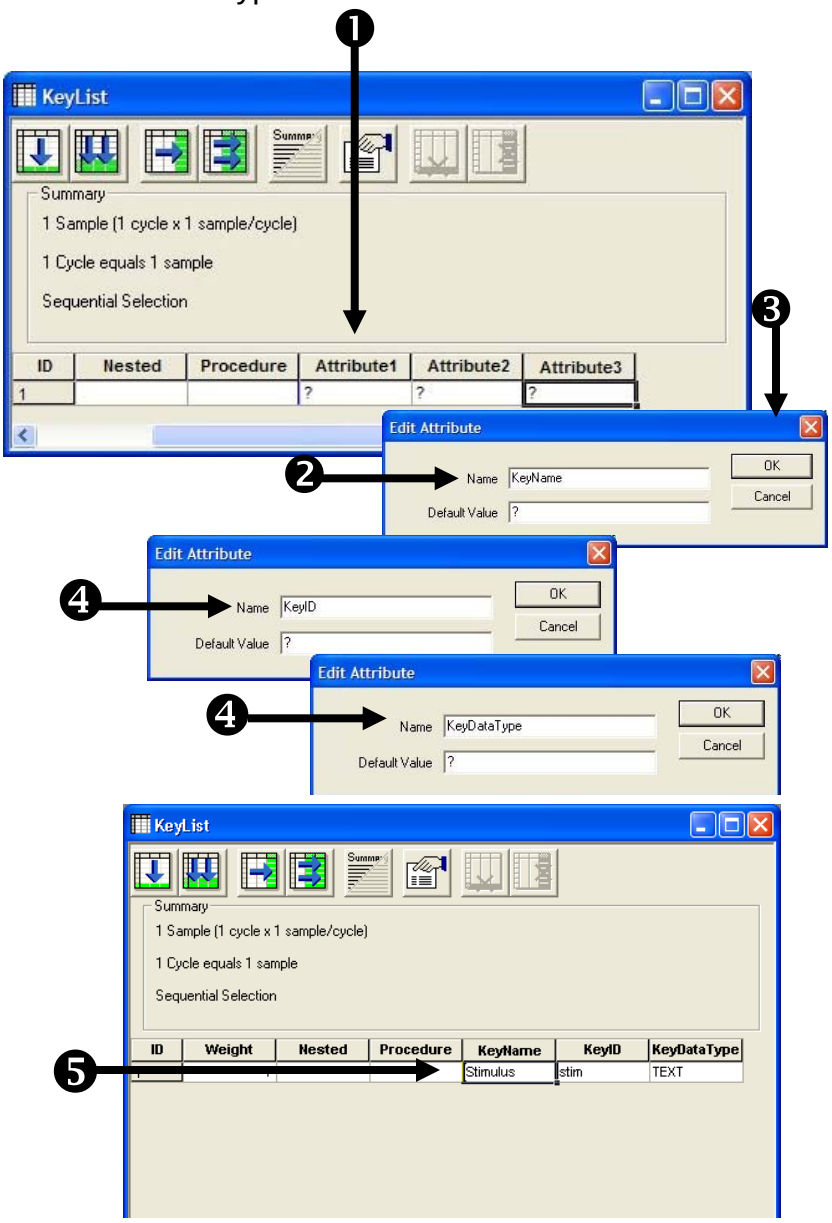
1) ***Double click*** the header of the **Attribute1** column to display the **Edit Attribute** dialog.

2) ***Type*** “**KeyName**” in the **Name** field of the **Edit Attribute** dialog.

3) ***Click*** the **OK** button to accept the change.

4) ***Repeat*** steps1-3 above to ***rename*** **Attribute2** and **Attribute3** to **KeyID** and **KeyDataType** respectively.

5) ***Click*** in each cell under the **KeyName, KeyID,** and **KeyDataType** attributes and ***edit*** the values to match those shown.



Task 3: Add a reference to the KeyList

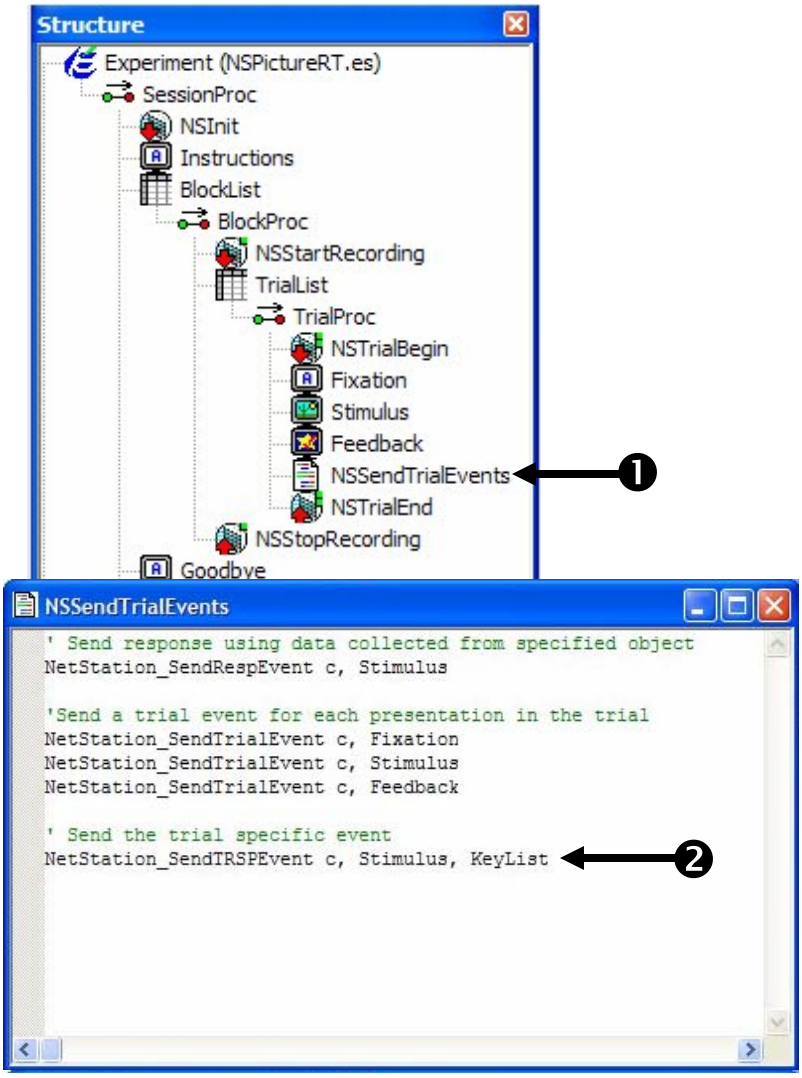
Add script to the NSSendTrialEvents Inline to append the KeyList to the package file routine that sends the TRial SPecific (TRSP) event to Net Station.

The NetStation_SendTRSPEvent includes an optional parameter that is used to specify a reference to a list of user defined keys. While it is acceptable to define multiple Key Lists within an experiment, typically only one list is used. Each Key List that is defined must be configured to fully describe the additional key data that is to be associated with the current trial.

Add the optional parameter to the end of the NetStation_SendTRSPEvent parameters list to associate the KeyList object with the TRSP event. At runtime the NetStation_SendTRSPEvent routine will process the KeyList by retrieving each KeyName, looking up its current value within the experiment Context, and then appending all of the key information to the end of the TRSP event. If a specified KeyName cannot be found in the context a runtime error will be generated.

1) ***Double click*** on the **NSSendTrialEvents** inline object to open the dialog box.

2) ***Type*** in the **script** exactly as shown to add a reference to the **KeyList** as the last parameter of the **NetStation_SendTRSP Event** routine.



Task 4: Run the experiment

Run the experiment to verify that the new key is being sent to Net Station and that no runtime errors are generated.

Your experiment should now be configured to send additional keys from the KeyList. Net Station enabled experiments can be run locally from E-Studio during development and testing (i.e. without having the Net Station hardware connected). You should fully test your experiment locally before deploying it to the experiment presentation computer connected to your Net Station hardware. Note that while many Net Station related errors can be detected while testing locally, some errors will only be detected or generated when the actual Net Station hardware is in use. Thus, you should fully test your experiment with the Net Station hardware connected and active prior to scheduling actual participants.

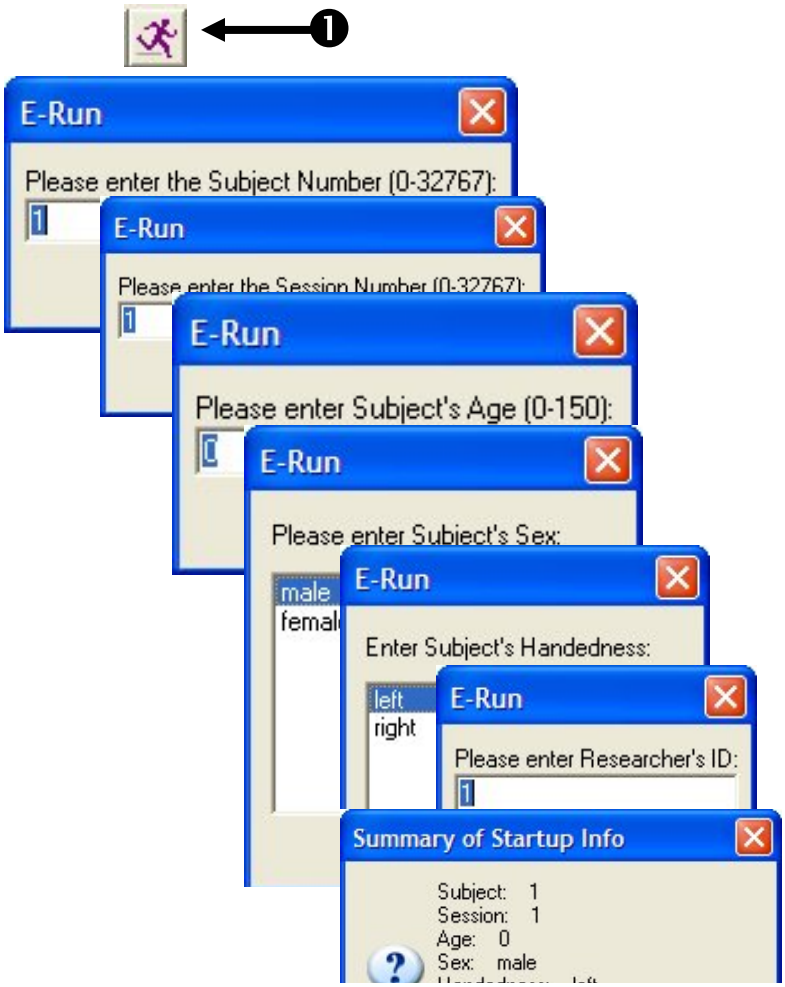
Once you have launched the experiment you will be presented with a series of initialization prompts. Press Enter or click OK to accept the default values during testing. Read and respond to any instructions presented then observe the stimulus presentation sequence to verify the experiment is functioning correctly and that no errors are generated. You should review the output in the Experiment Status Control Panel within the Net Station Workbench interface to verify that the TRSP event is displaying the appropriate keys and key values on each trial.

1) ***Click*** the **Run icon** on the application toolbar to generate and run the experiment locally.

2) ***Press* Enter** to accept the default values for each of the initialization prompts presented.

3) ***Observe*** the stimulus presentation sequence to verify the experiment is functioning correctly and completes with no errors being generated.

4) ***Save*** the experiment.



Tutorial 3 – Adding a practice block

Summary: It is common for the design and implementation of an experiment to evolve over time during its useful life cycle (e.g. number of trials are increased/decreased as new stimuli are added/deleted, new experimental conditions/cells are introduced, participant friendly breaks are introduced, etc.) Many experiments introduce blocks of practice trials after a basic version of the experiment is functioning correctly. Within a Net Station enabled experiment, introducing a block of practice trials often theoretically introduces a new set of cells that need to be reflected to Net Station (e.g. practice related data can be ignored during critical data analysis). This process typically involves adding or restructuring entries in the Cell List and assigning new Cell Numbers to the appropriate trials.

Goal: This tutorial illustrates how to add a new block of practice trials into the NSPictureRT sample experiment created during *Tutorial 1* and *Tutorial 2*. You will edit the experiment's Cell List and assign new Cell Numbers and Cell Labels to the trials of the practice block.

Overview of Tasks:

- Add two new cells/rows to the CellList
- Add a new PracticeBlockProc procedure to the BlockList object
- Add the required Net Station package calls to the PracticeBlockProc
- Add and configure a new PracticeTrialList object to represent the new practice trials
- Add instruction displays to begin and end the block of practice trials
- Verify the overall experiment structure and run the experiment

Recommended readings:

It is required that you have read through and completed at least *Tutorial 1* before beginning this tutorial. It is recommended that you have also completed *Tutorial 2*.

Estimated Tutorial Time: 15-30 minutes.

Task 1: Add new cells to the CellList

Add two new cells to the CellList then edit the attribute values to define the new practice cells used within the experiment.

Adding a block of practice trials to the experiment will conceptually create two new cells (which will be name “redcar_prac” and “bluecar_prac” in this example). Add two new rows to the CellList and assign the CellNumber and CellLabel attributes accordingly to define the two new cells. The values for the CellNumber attribute must be numeric and must be unique within the experiment. You may currently define up to 255 unique cells within an experiment. It is recommended that when you add new cells to the CellList you assign new CellNumbers sequentially (e.g. to assure compatibility with any previously collected data). Values for the CellLabel can be any string you choose to identify and describe the cell (i.e. CellLabels do not have to be four character codes).

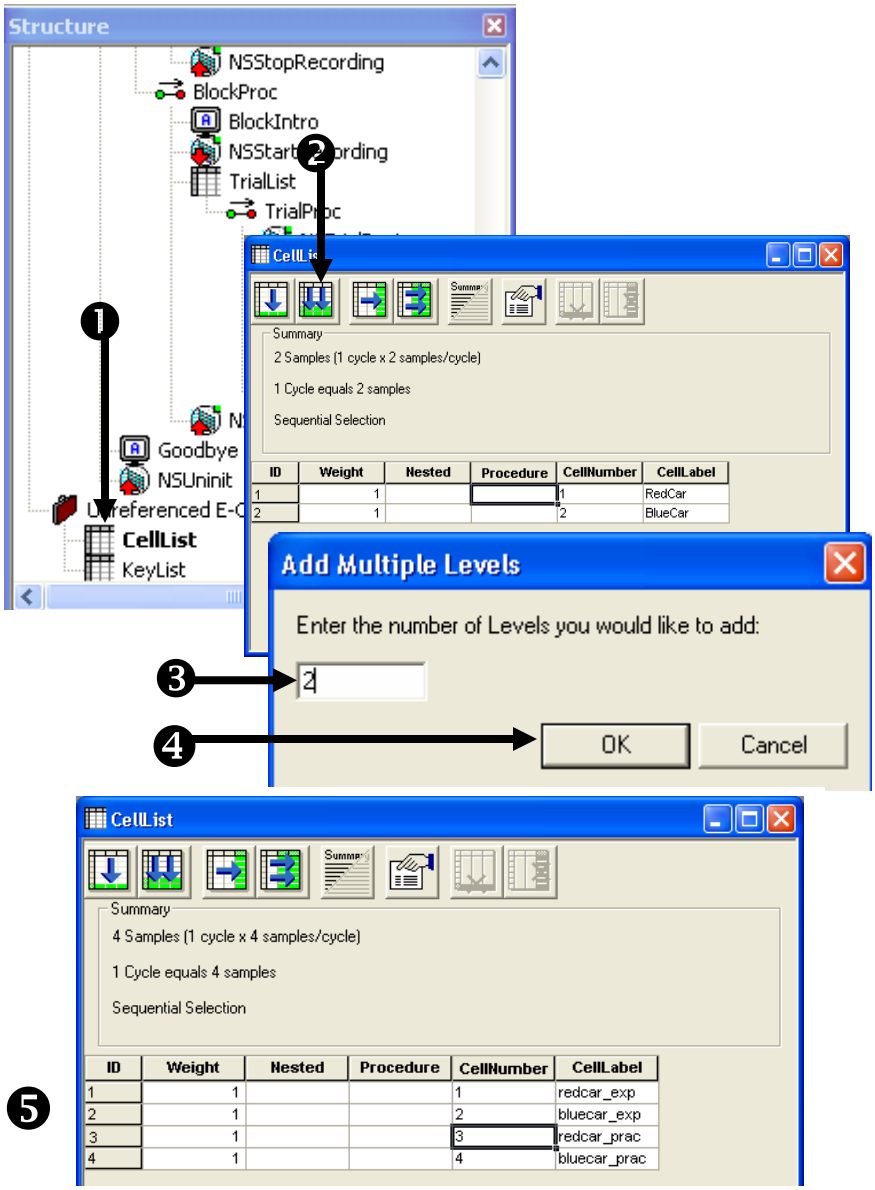
1) ***Open*** **NSPictureRT.es** and ***double click*** the **CellList** object to open it in the workspace.

2) ***Click*** the **Add Multiple Levels** button.

3) ***Type*** **2** in the **Add Multiple Levels** dialog.

4) ***Click*** the **OK** button to add two new levels/rows to the **CellList**.

5) ***Click*** in each cell under the **CellNumber** and **CellLabel** attributes and ***edit*** the values to match those shown.



Task 2: Edit the BlockList to add a PracticeBlockProc Procedure

Add a new row to the BlockList object and create a new PracticeBlockProc Procedure to represent the practice block.

Create a new level/row in the BlockList to represent the new block of practice trials. Type "PracticeBlockProc" under the Procedure attribute to create a new Procedure object that will be used to execute the new block. Allow E-Studio to create the new PracticeBlockProc object and set the existing PracticeMode attribute to "Yes" to log an indication that this block is a block of practice trials.

1) ***Double click*** the **BlockList** object to open it in the workspace.

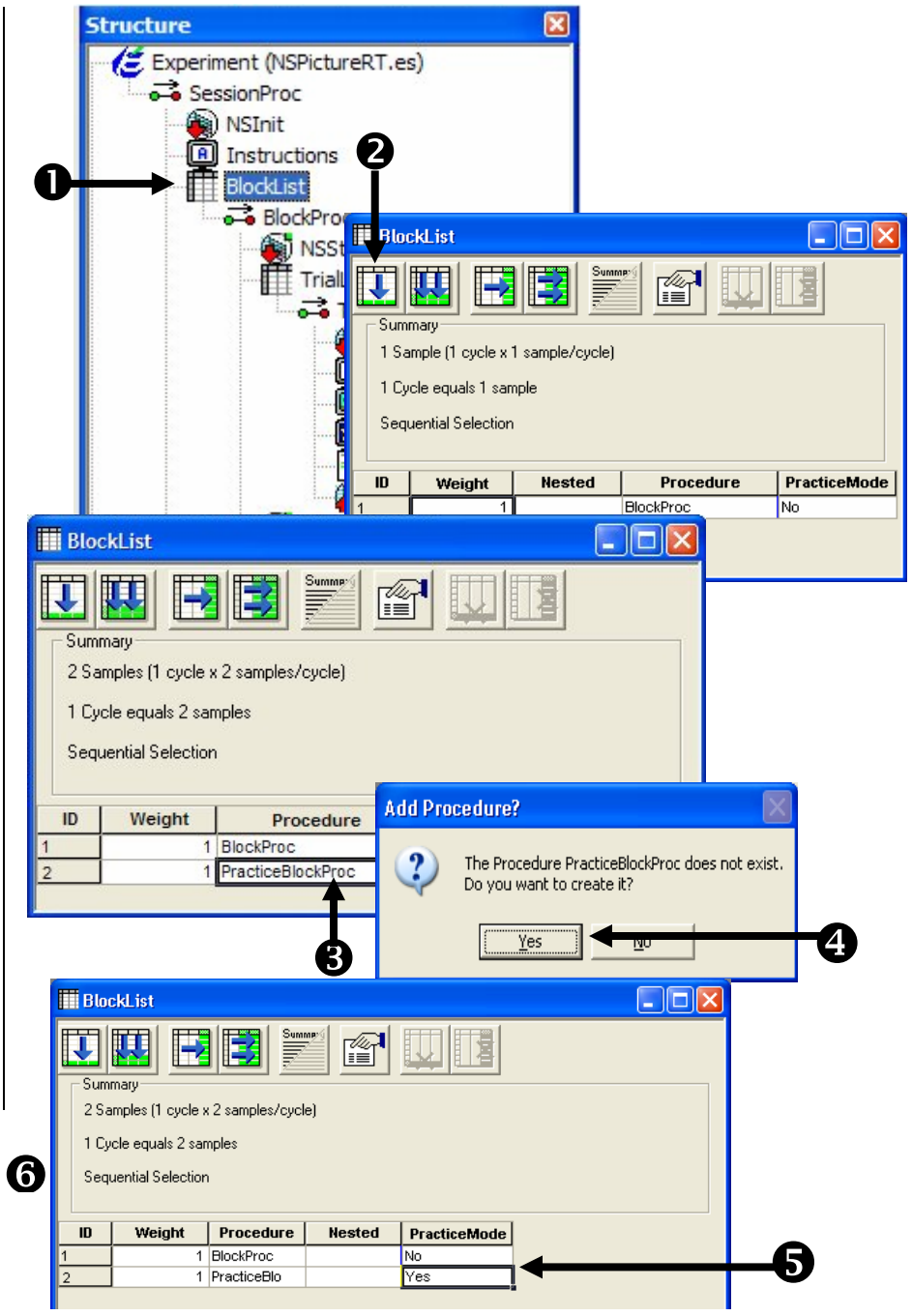
2) ***Click*** the **Add Level** button to add a new leve/row to the **BlockList.**

3) ***Click*** in the cell under the **Procedure** attribute in the second row, **type PracticeBlockProc,** and then ***press*** **Enter**.

4) ***Click*** the **Yes** button to create a new **PracticeBlockProc** Procedure in the experiment.

5) ***Click*** in the cell under the **PracticeMode** attribute in the second row, ***type*** **Yes**, and ***press*** **Enter**.

6) ***Verify*** that your **BlockList** matches the picture.



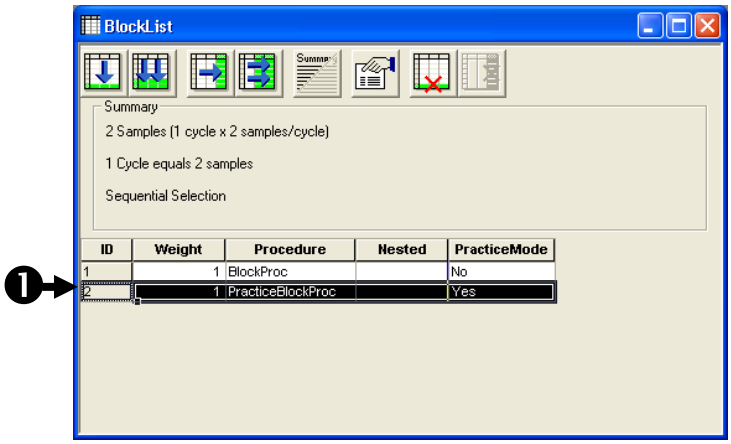
Task 2 (continued): Edit the BlockList to add a PracticeBlockProc Procedure

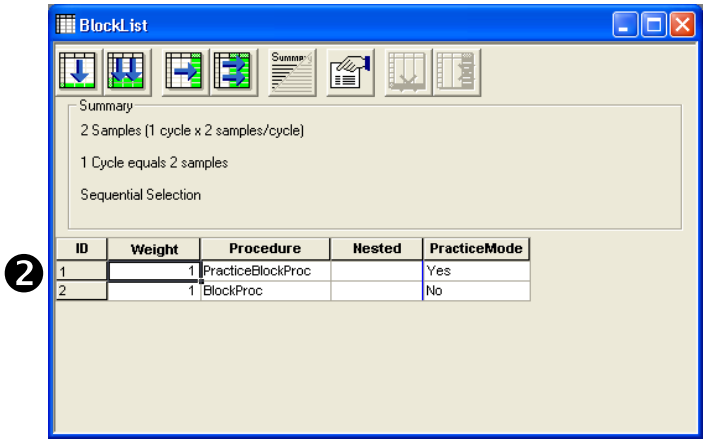
Reorder the levels/rows of the BlockList so that the practice block is run first.

When a List object is configured to select exemplars sequentially, the order of the rows directly determines the sampling order. Click the row header for the second row and drag it above the first row so that the block of practice is run first followed by the block of real trials. If you want to run multiple blocks of practice or real trials you can edit the cells under Weight attribute as desired.

1) ***Click*** the **ID** row header to select the **second row**.

2) ***Drag*** the **second row** above the **first row** to ***reorder*** the blocks then ***verify*** that your **BlockList** matches the picture.
During the drag operation you will see a red line indicating where the selected row will be moved to when it is dropped.





Task 3: Specify the PracticeBlockProc

Add the required Net Station package calls to the beginning and ending of the PracticeBlockProc.

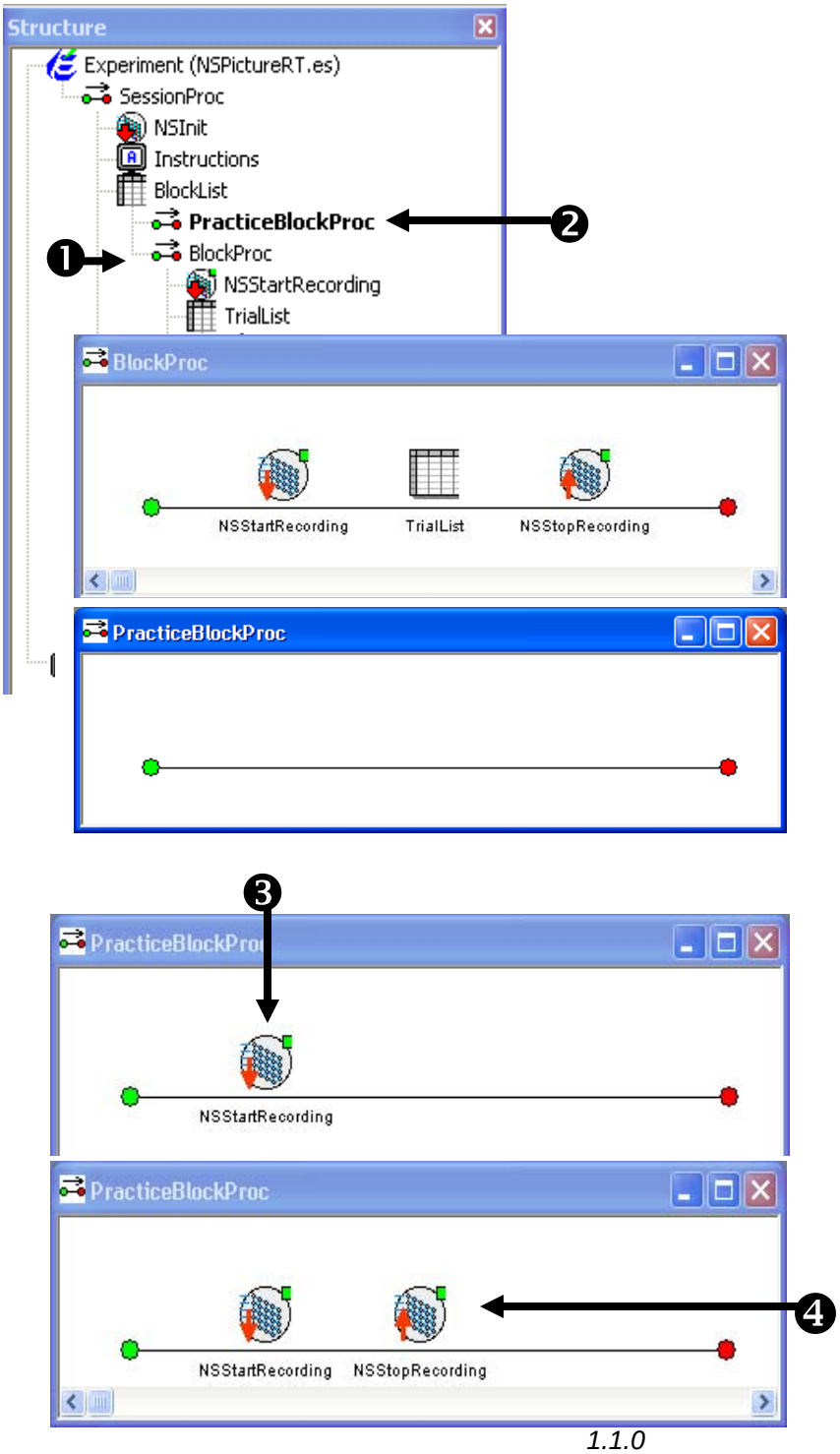
Configure the PracticeBlockProc to start Net Station recording at the beginning of the practice block and stop Net Station recording at the end of the practice block. You can reuse the NSStartRecording and NSStopRecording objects that are already being called from the original BlockProc by dragging and dropping them on PracticeBlockProc object.

1) ***Double click*** the **BlockProc** object to open it in the workspace.

2) ***Double click*** the **PracticeBlockProc** object to open it in the workspace.

3) ***Drag*** the **NSStartRecording** object from the **BlockProc** and ***drop*** it in the **PracticeBlockProc**.

4) ***Drag*** the **NSStopRecording** object from the **BlockProc** and ***drop*** it as the last object in the **PracticeBlockProc**.



Task 4: Create and configure the PracticeTrialList

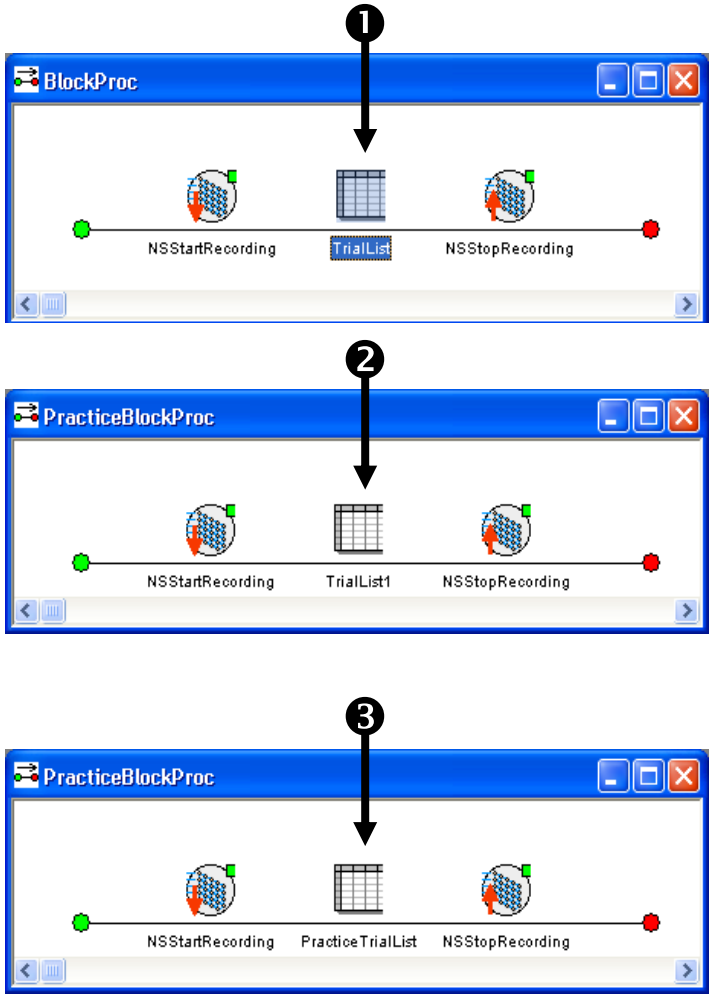
Create a new List to represent the practice trials by making a copy of the TrialList object.

The new block of practice trials will require a unique List at the trial level since it will need to reference CellNumbers that are different than those used in the real trials. Since the existing TrialList object is already configured appropriately for use with Net Station you can copy it and rename it to quickly create a PracticeTrialList for use in the PracticeBlockProc. To make a copy of an existing object you select the object and then drag it into position while holding down the Ctrl key. When a copy operation is in progress the mouse cursor will include a plus sign (+). Be sure to keep the Ctrl key held down until after you release the mouse button.

1) ***Click*** the **TrialList** object to select it within the **BlockProc**.

2) ***Hold down*** the **Ctrl** key, ***drag*** the **TrialList** object over the **PracticeBlockProc** and ***drop*** it between the **NSStartRecording** and **NSStopRecording** objects.
Dragging an object while holding down the Ctrl key will make a new copy of the object. The mouse cursor will include a plus sign (+) when a copy operation will occur. Notice that the new object that is created is named TrialList1.

3) ***Click*** on the **TrialList1** object to select it then ***press*** **F2** and ***rename*** the object as **"PracticeTrialList"**.

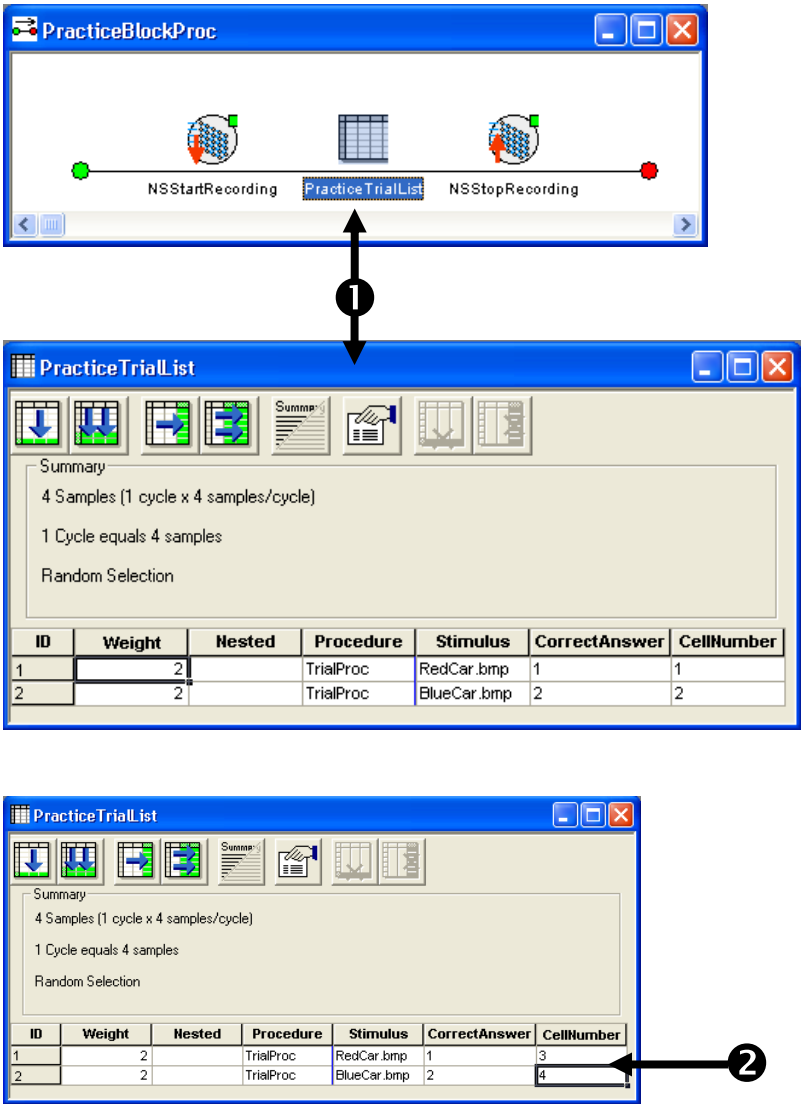


Task 4 (continued): Create and configure the PracticeTrialList
Assign the CellNumbers required for the practice trials.

Because the PracticeTrialList was created by copying the existing TrialList the CellNumbers are incorrect for use in the practice block. The practice trials must reference the new cells that were created in the CellList (i.e., "redcar_prac" and "bluecar_prac"). Edit the cells under the CellNumber attribute to reference the new cell numbers of 3 and 4 respectively.

1) ***Double click*** the **PracticeTrialList** object open it in the workspace.

2) ***Click*** in the cells under the **CellNumber** attribute and ***edit*** the **values** to match those in the picture.



Task 5: Add an instruction display to begin practice

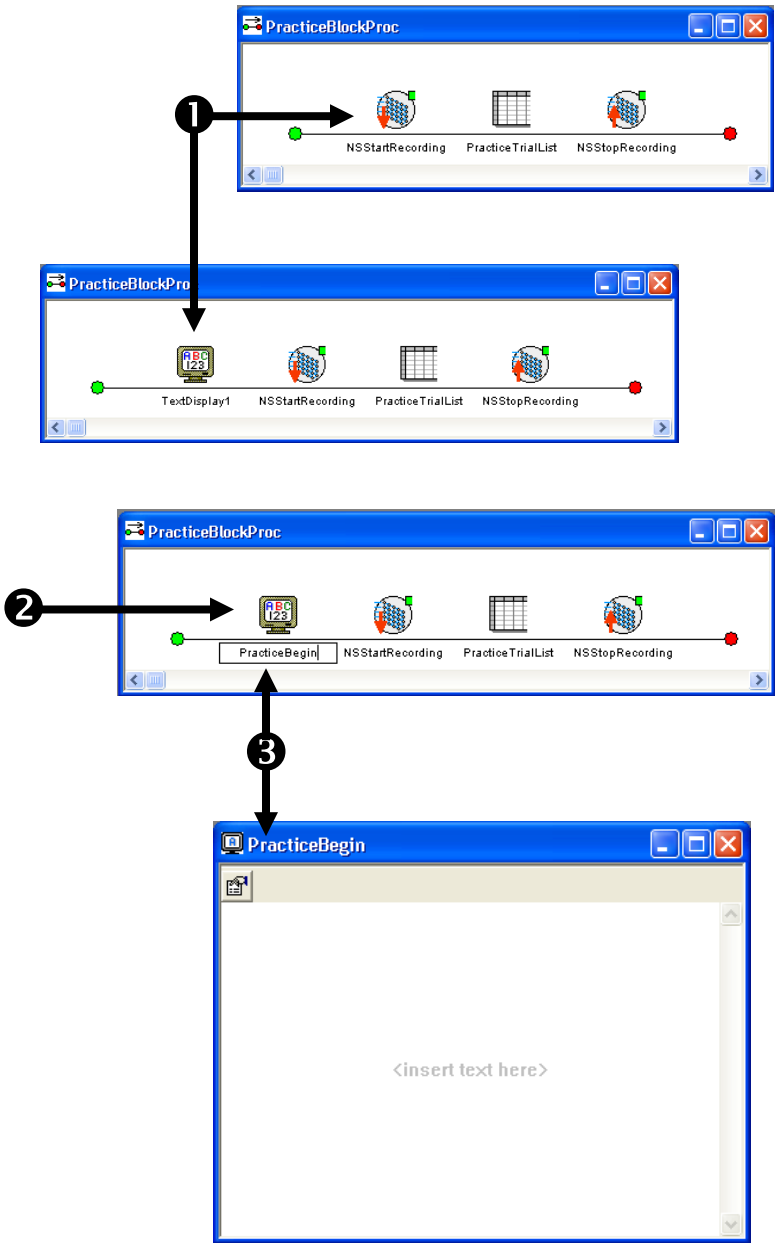
Add a new TextDisplay at the beginning of the practice block to orient the participant.

Create a new TextDisplay object at the very beginning of the PracticeBlockProc (before Net Station recording begins) to notify the participant that they are about to begin a block of practice trials. Note, if you want to combine text, images, and sound to present instructions within your own experiments it is recommended that you use a Slide object instead of a TextDisplay object.

1) ***Drag*** a new **TextDisplay** object from the **Toolbox** and ***drop*** it at the beginning of the **PracticeBlockProc**.

2) ***Click*** on the **TextDisplay1** object to select it then ***press* F2** and ***rename*** the object as **"PracticeBegin"**.

3) ***Double click*** the **PracticeBegin** object to open it in the workspace.



Task 5 (continued): Add an instruction display to begin practice

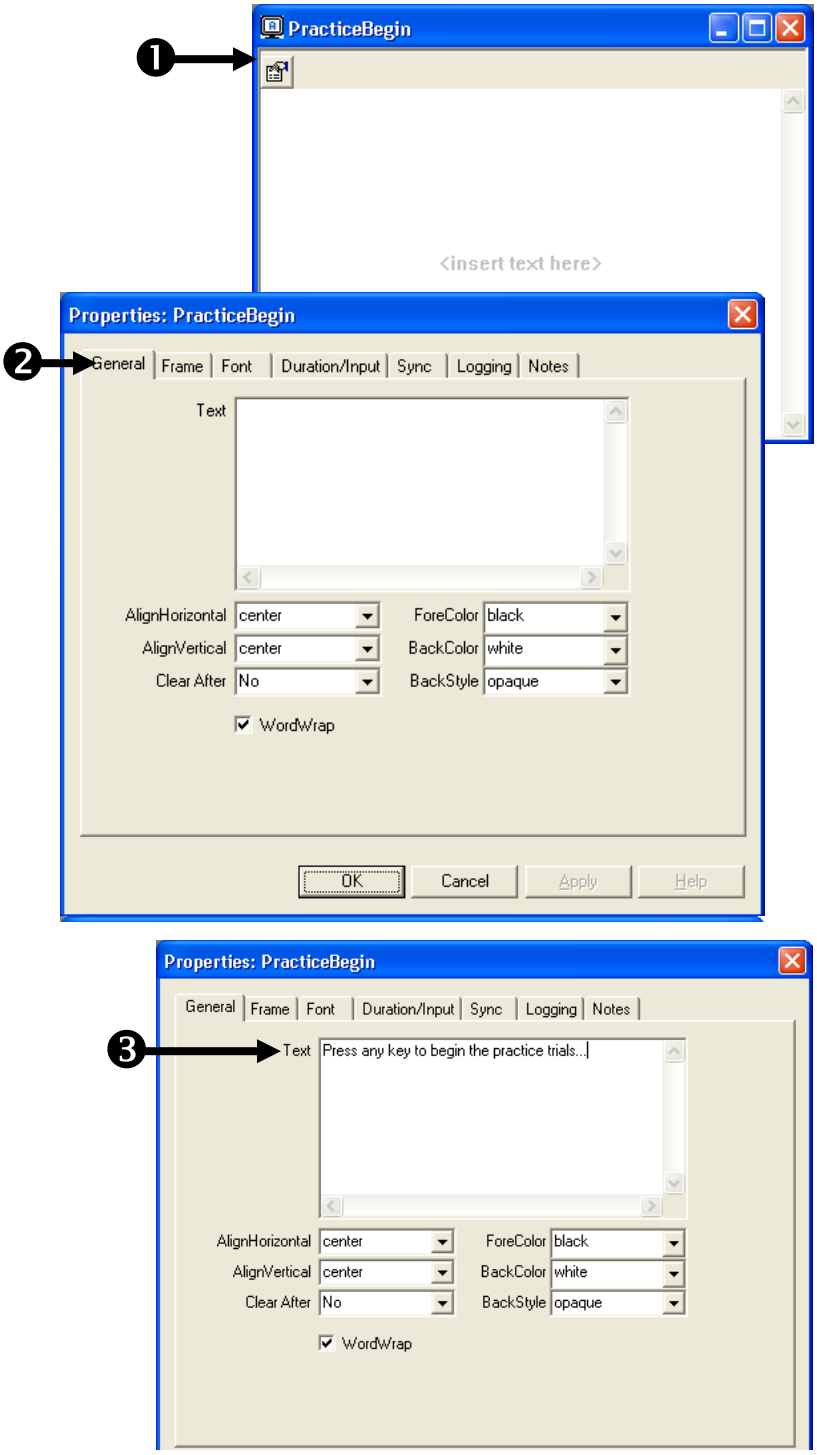
Type the text that you want to present to the participant.

Open the Properties dialog for the PracticeBegin object and type in the text that you want to have presented to the participant prior to the start of the practice block.

1) ***Click*** the **Properties** button to display the **Properties dialog** for the **PracticeBegin** object.

2) ***Click*** on the **General** tab to select it.

3) ***Click*** in the **Text** field and ***type*** "**Press any key to begin the practice trials...**".



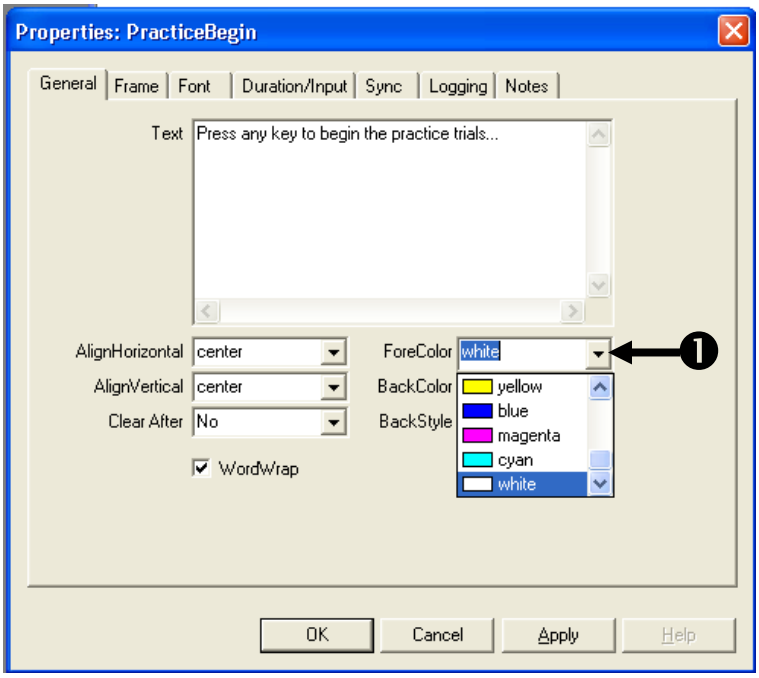
Task 5 (continued): Add an instruction display to begin practice
Set the foreground and background colors for the instruction display.

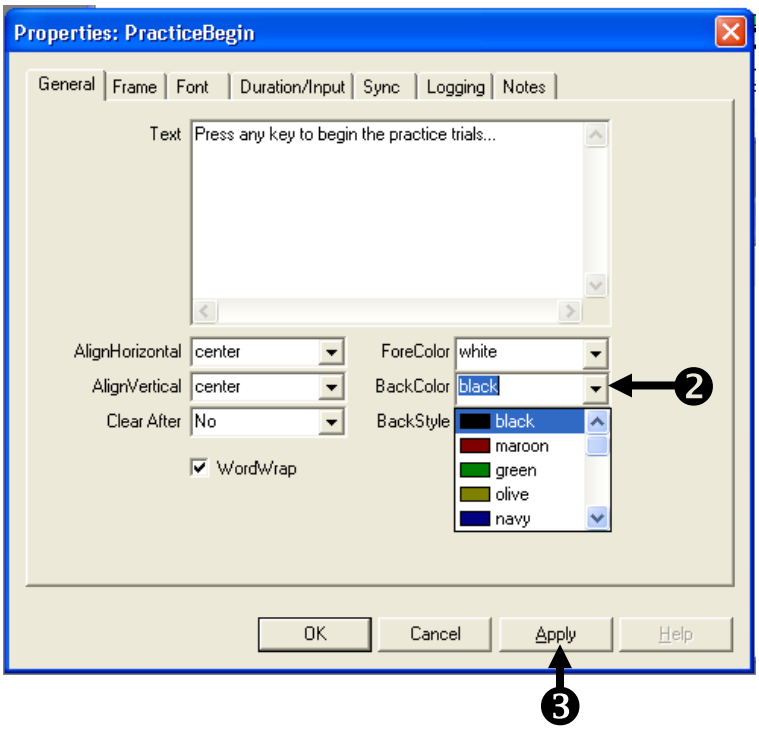
The PracticeBegin object is currently configured to present black text on a white background however other displays within the experiment are configured to present white text on a black background. Change the foreground and background properties of the PracticeBegin object to match those that are used in the rest of the experiment.

1) ***Click*** the **dropdown** beside the **ForeColor** field and ***select*** "**white**" as the text foreground color.

2) ***Click*** the **dropdown** beside the **BackColor** field and ***select*** "**black**" as the text background color.

3) ***Click*** the **Apply** button to save the changes to this tab.





Task 5 (continued): Add an instruction display to begin practice

Configure the PracticeBegin object to wait until the participant presses a key to continue.

Set the properties of the PracticeBegin object so that it presents the instructions to the participant and then waits indefinately until the a key is pressed to terminate the display and begin the block of practice trials. Specifically, set the Duration property to "(infinite)" and then add a new Input Mask which references the Keyboard device. The default settings for the various Response Options are acceptable and do not need to be edited further.

1) ***Click*** the **Duration/Input** tab to select it.

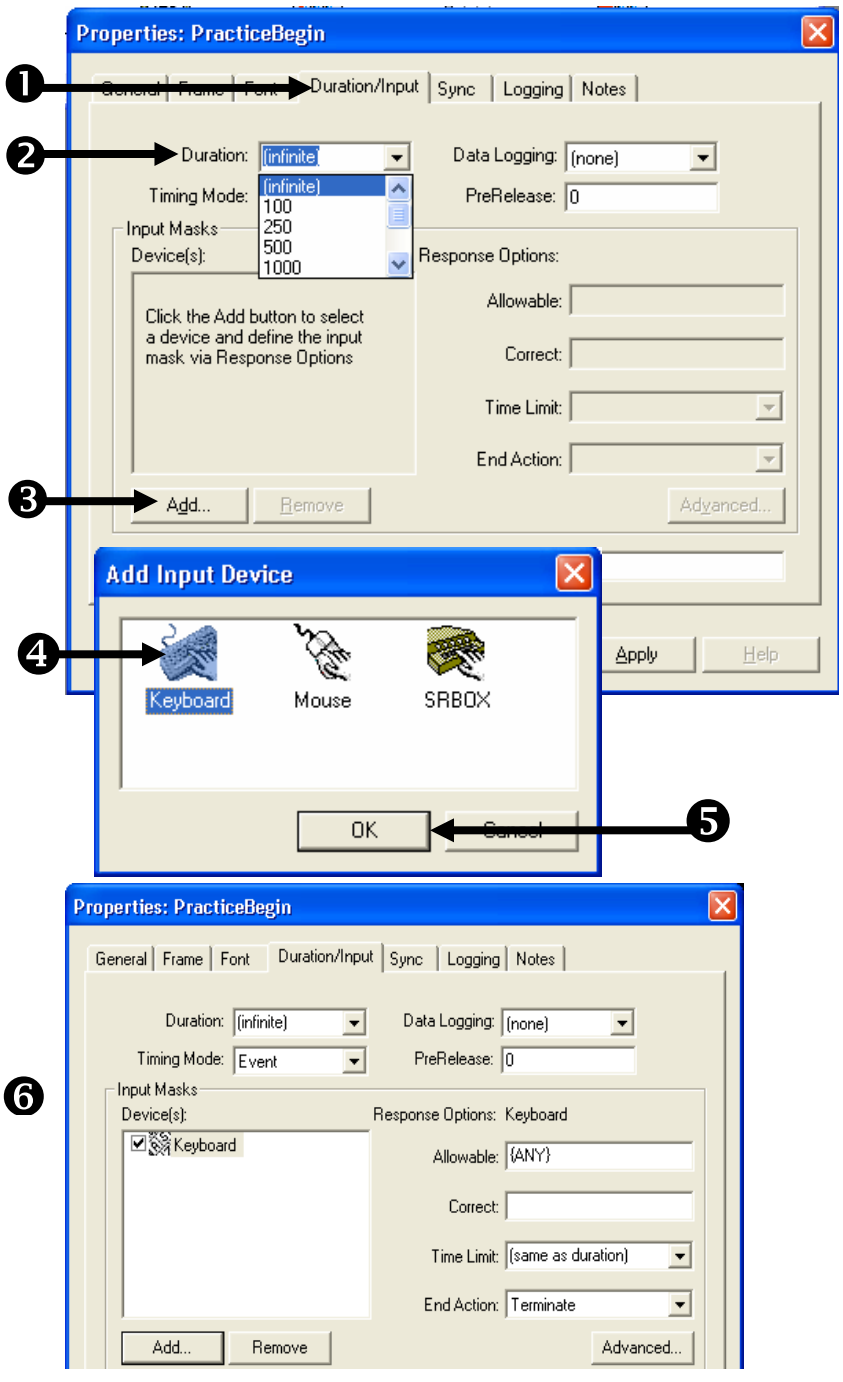
2) ***Click*** the **dropdown** beside the **Duration** field ***select*** "**(infinite)**" as the duration of the object.

3) ***Click*** the **Add...** button.

4) ***Click*** on the **Keyboard** device to select it.

5) ***Click*** the **OK** button to add a new input mask for the **Keyboard** device.
By default the input mask will allow any key press to terminate the input.

6) ***Verify*** that your **Duration/Input** tab matches the picture then ***click*** the **OK** button to accept all changes.



Task 6: Add an instruction display to end practice

Add a new TextDisplay at the end of the practice block to notify the participant that the practice trials are complete.

Create a new TextDisplay object to notify the participant that the practice trials have been completed and the real trials are about to begin. Since the PracticeBegin object is already configured appropriately for use in the experiment (e.g. foreground color, background color, wait for a key press to continue) it can be copied, renamed and edited for this purpose. Click on the PracticeBegin object, hold down the Ctrl key and copy it to the very end of the PracticeBlockProc. Rename the new object as "PracticeEnd" and then open it and edit the text message it presents to appropriately notify the subject.

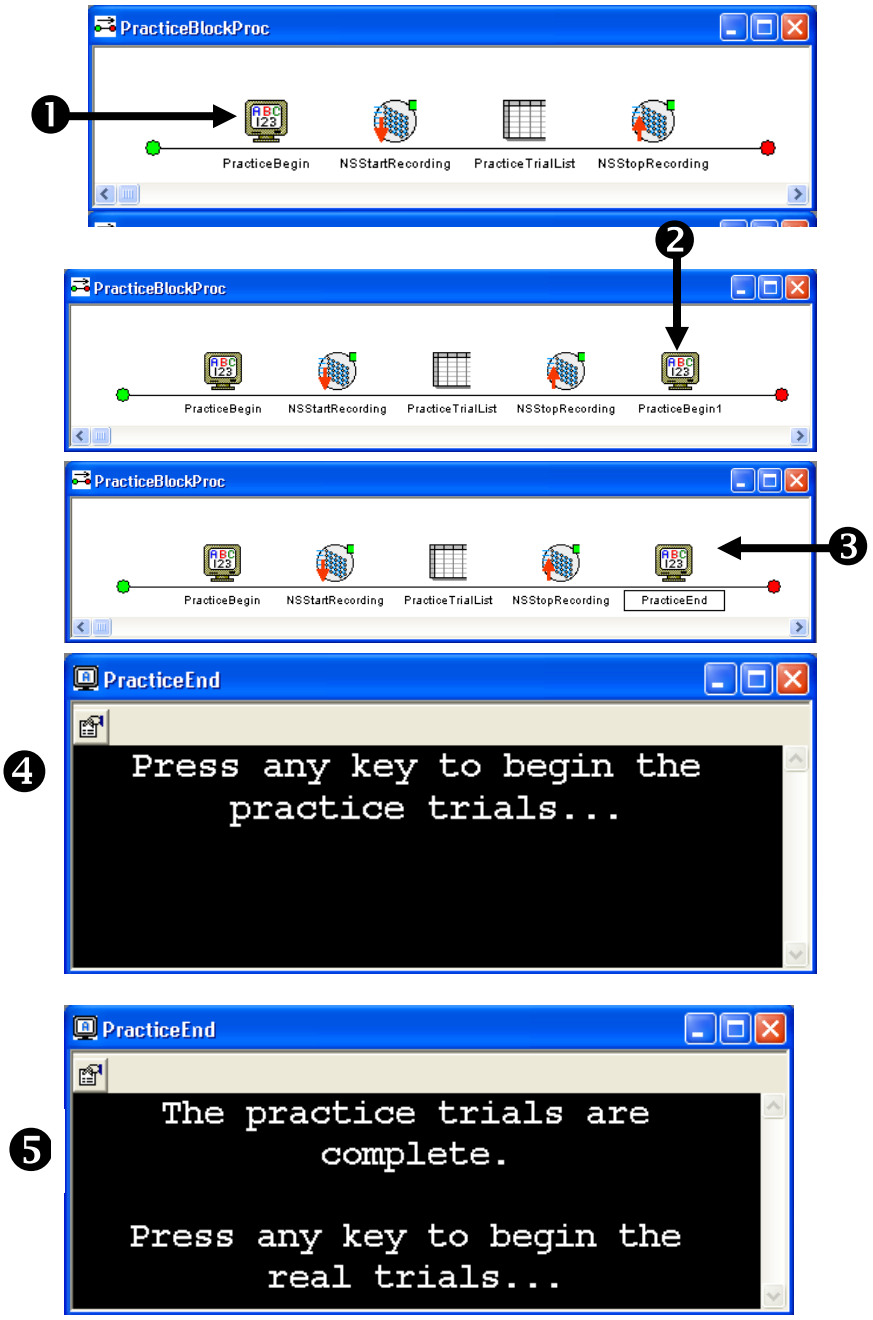
1) ***Click*** the **PracticeBegin** object to select it within the **PracticeBlockProc**.

2) ***Hold down*** the **Ctrl** key, ***drag*** the **PracticeBegin** and ***drop*** it at the end of the **PracticeBlockProc**. *Notice that the new object is named PracticeBegin1.*

3) ***Click*** on the **PracticeBegin1** object to select it then ***press*** **F2** and ***rename*** the object as **"PracticeEnd"**.

4) ***Double click*** the **PracticeEnd** object to open it in the workspace.

5) ***Edit*** the **text** to match the picture.



Task 7: Run the experiment

Run the experiment to verify that the appropriate prompts are presented and no runtime errors are generated.

Your experiment should now be designed to present a block of practice trials prior to beginning the real trials. Net Station enabled experiments can be run locally from E-Studio during development and testing (i.e. without having the Net Station hardware connected). You should fully test your experiment locally before deploying it to the experiment presentation computer connected to your Net Station hardware. Note that while many Net Station related errors can be detected while testing locally, some errors will only be detected or generated when the actual Net Station hardware is in use. Thus, you should fully test your experiment with the Net Station hardware connected and active prior to scheduling actual participants.

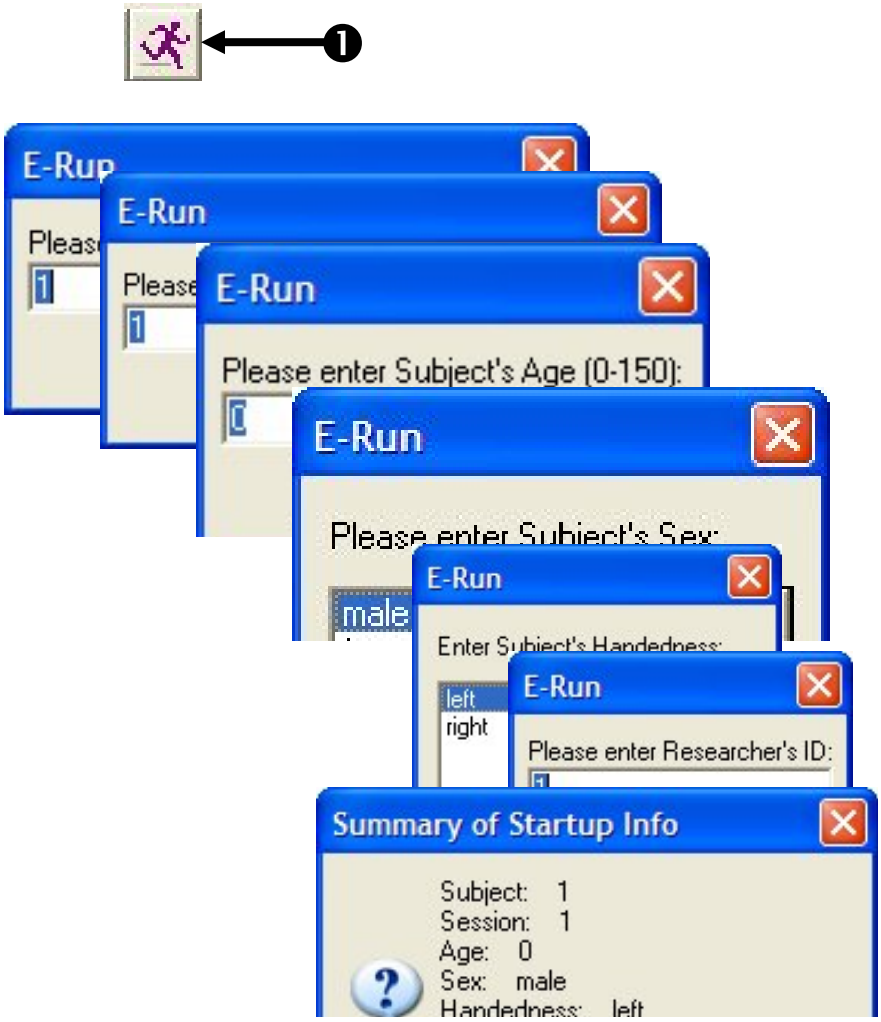
Once you have launched the experiment you will be presented with a series of initialization prompts. Press Enter or click OK to accept the default values during testing. Read and respond to any instructions presented then observe the stimulus presentation sequence to verify the experiment is functioning correctly and that no errors are generated.

1) ***Click*** the **Run icon** on the application toolbar to generate and run the experiment locally.

2) ***Press* Enter** to accept the default values for each of the initialization prompts presented.

3) ***Observe*** the stimulus presentation sequence to verify the experiment is functioning correctly and completes with no errors being generated.

4) ***Save*** the experiment.



Chapter 3: Reference Guide

The Single Clock

The Single Clock timing method

Although the clock on your PC and the clock on the amplifier are very precise, every clock has some very small, unique fraction of error in its designated frequency. The difference in error rates can cause the PC and amplifier clocks to drift apart by several hundreds of milliseconds per hour, resulting in corresponding drift between E-Prime and Net Station timestamps. Without correction, this drift can be quite significant over the course of an experiment.

The Single Clock timing method enables E-Prime to obtain time from the amplifier clock, just as Net Station does. With a shared clock, drift is eliminated, ensuring accurate timing in your experiments

How the Single Clock works

The Net Amps 200 and Neurotravel USB amplifiers are each designed to output a 1 KHz TTL timing signal pulse, which is generated from the amplifier's internal clock chip. The Net Amps 200 outputs this signal through the Experimental Synch port on its rear panel, while the Neurotravel amplifier uses the Trigger Out port on its front panel. The timing signal pulse is transmitted from the amplifier port to the EGI Single Clock Experimental Control Timing Box through a custom EGI Timing Cable.

The Single Clock Experimental Control Timing Box converts the timing signal pulse from the amplifier into a timestamp, which is relayed to E-Prime through the National Instruments ribbon cable and DIO96 PCI card. When the amplifier is turned off in Net Station, it stops outputting a timing signal pulse, and the timing box resets its counter to 00:00. When the amplifier is turned on again in Net Station, the timing signal pulse resumes, and both Net Station and the timing box begin counting from 00:00. In this way, the timestamps in Net Station and E-Prime are always the same.

Timing Test

The Single Clock timing method is very efficient and reliable, but it is always prudent to verify that your new system is functioning within specification before

you begin collecting any real data. Running the sample *NSTimingTest.es* experiment provided with the *E-Prime Extensions for Net Station installation* will help you check your system's timing accuracy. You should also repeat this process to verify timing in each new experiment you create, prior to running subjects.

Once you record test data using the *NSTimingTest.es* experiment and your EGI AV Tester, you can review the timing results either directly in the Net Station file, or by using the EGI Event Timing Tester application provided with Net Station. Your method of analysis will vary, depending on which amplifier you are using.

The Net Amps 200 has a DB9 Digital Input (DIN) port on its back panel, which allows DIN events output from the AV Tester device to be recorded in the Net Station file. The Neurotravel amplifier does not currently have a DIN port, so you must use the 400 microvolt output on the AV Tester to send a square wave signal directly to one of the amplifier's channel inputs. The square wave can then be compared to the events in the recording file.

Please refer to the Net Station, System, and AV Tester manuals that came with your EGI system for more information on how to use those EGI products, or contact EGI Support. If you have questions regarding how to set up and run experiments in E-Prime, refer to *Chapter 1* of this manual, "*Getting Started Guide*".

Run the NSTimingTest.es sample experiment:

1. Once you have set up your AV Tester equipment and amplifier according to the EGI manuals, launch an appropriate Net Station Acquisition Setup on the Macintosh. You should have the Multi-Port ECI Log and device panels open. Use default port 55513.

2. If you a using the Net Amps 200, make sure that you have enabled *only* DIN 1 in the Net Station Digital Inputs panel. This will prevent signal bleed over and provide a clean DIN event in the recording file.

Note: The four AV Tester DIN outputs differ in sensitivity. Under most circumstances, DIN1 is appropriate for visual stimuli, while DIN 2 is better for auditory stimuli. Some monitors, such as flat panels, may require using DIN 3. Some experimentation may be necessary.

3. Turn on the Workbench to start up the amplifier.

4. If you are using the Neurotravel amplifier, you will have already set up the AV Tester so that it is outputting a 400 microvolt signal into one of the amplifier channel ports on the front of the amp. You should now see that flat line signal on

the selected channel in the Net Station Dense Waveform Display. Make sure that the signal is clearly defined and free of excess noise in the display.

3. On the PC, launch the E-Prime sample experiment, “NSTimingTest.es”. The default location is:

My Documents\My Experiments\Net Station\Samples\NSTimingTest.

4. Specify “socket” and your Macintosh computer’s IP address (“XXX.XXX.XXX.XXX”) in the NSInit parameters.

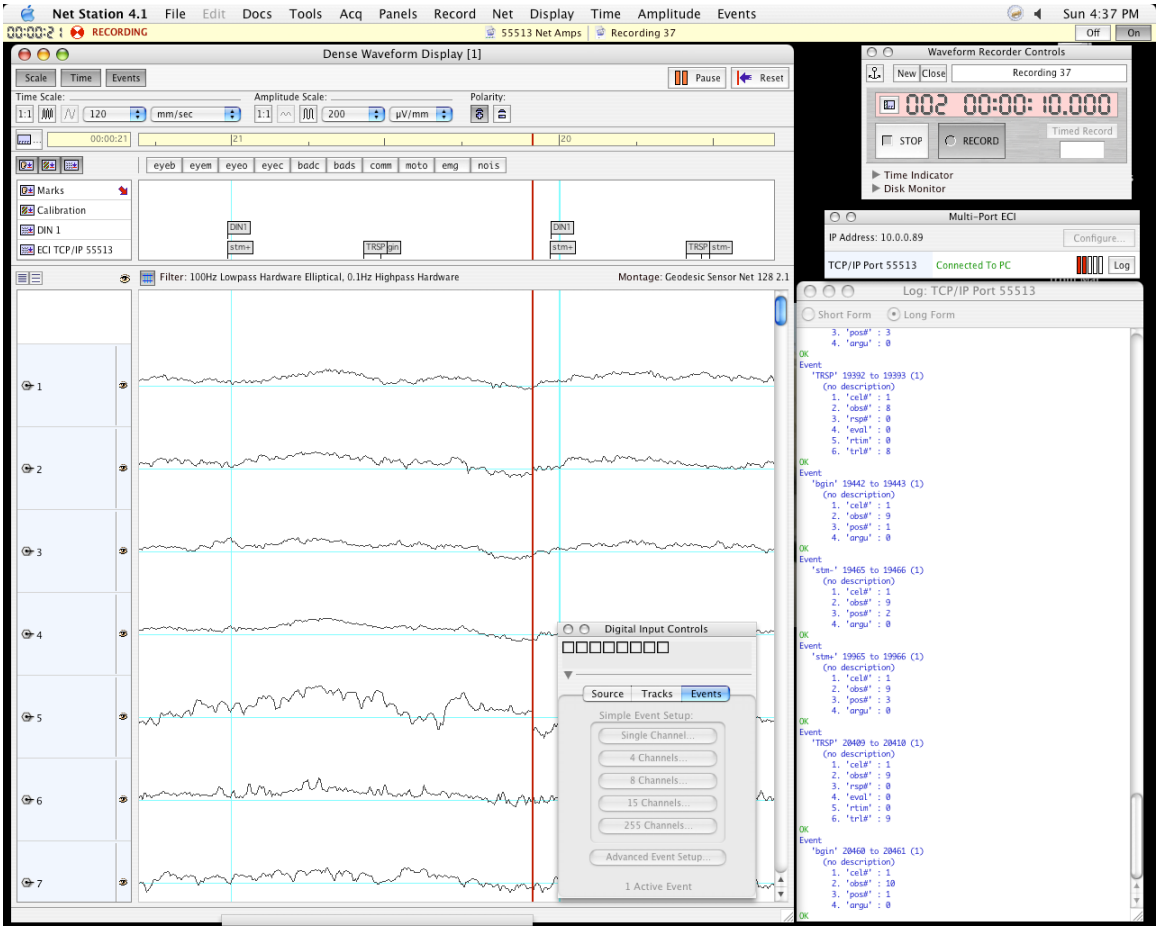
5. Select E-Run> Run from the application menu (or click the Run button) to generate and begin running the experiment.

6. After the entering StartUp Info in the message boxes, line up the AV Tester Optical Sensor with the white Stimulus square. Make sure that the sensor is even and as close as possible to the monitor screen.

7. Continue running the experiment. You can monitor events in the Net Station Multi-Port ECI log and Dense Waveform Display Event Track as the experiment progresses.

8. The NSTimingTest.es experiment flashes a white Stimulus rectangle on the screen every second for one hour. Each Stimulus flash corresponds with a “stim+” event.

With the Net Amps 200, the AV Tester causes DIN events to be placed in the recording each time a flash of the Stimulus rectangle occurs. With the Neurotravel amplifier, the AV Tester outputs a square wave for each Stimulus flash.



7. After you have successfully run NSTimingTest.es and have reviewed the resulting Net Station recording, close the file and quit Net Station.

How to review a DIN file with Event Timing Tester

If you used the Net Amps 200 for your testing and recorded DIN events in your file, you can use the EGI Event Timing Tester application to review your timing results. If you used the Neurotravel amplifier, skip to the next section.

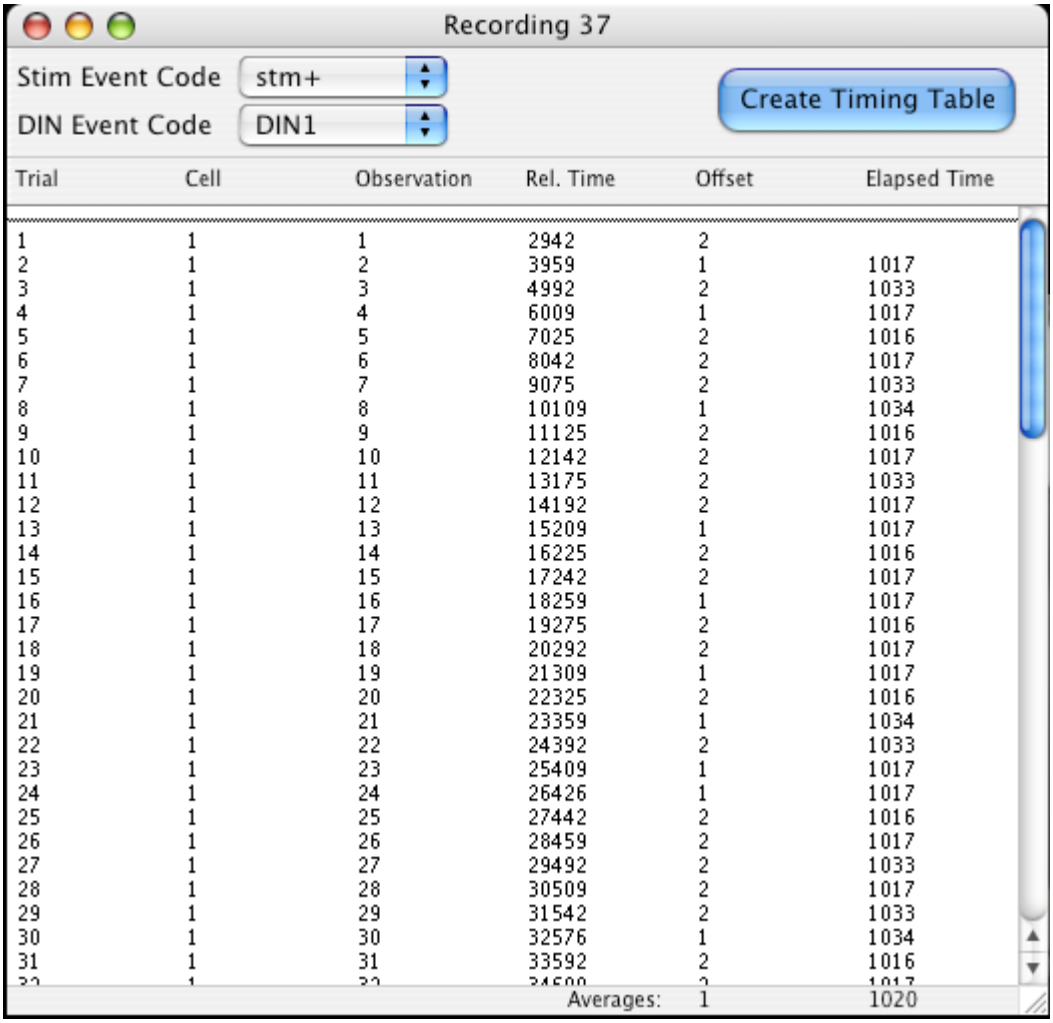
1. On the Net Station Macintosh, Launch the EGI Event Timing Tester (ETT) application. The default file path is: Applications> Net Station> Extras> Event Timing Tester.

2. Choose “Open” from the ETT File menu, and select your timing test recording using the file browser dialog. ETT will load event information from your file and present a File panel.

3. From the “Stim Event Code” pop-up menu, select “stm+”. This is the stimulus event from the NSTimingTest.es experiment.

4. From the “DIN Event Code” pop-up menu, select “DIN1”, or whichever DIN code you used in Net Station.

5. Click the “Create Timing Table” button to generate a table of event information. Each row represents a single trial (flash of the Stimulus rectangle) from the NSTimingTest.es experiment. The columns contain specific details about each trial.



Trial	Cell	Observation	Rel. Time	Offset	Elapsed Time
1	1	1	2942	2	
2	1	2	3959	1	1017
3	1	3	4992	2	1033
4	1	4	6009	1	1017
5	1	5	7025	2	1016
6	1	6	8042	2	1017
7	1	7	9075	2	1033
8	1	8	10109	1	1034
9	1	9	11125	2	1016
10	1	10	12142	2	1017
11	1	11	13175	2	1033
12	1	12	14192	2	1017
13	1	13	15209	1	1017
14	1	14	16225	2	1016
15	1	15	17242	2	1017
16	1	16	18259	1	1017
17	1	17	19275	2	1016
18	1	18	20292	2	1017
19	1	19	21309	1	1017
20	1	20	22325	2	1016
21	1	21	23359	1	1034
22	1	22	24392	2	1033
23	1	23	25409	1	1017
24	1	24	26426	1	1017
25	1	25	27442	2	1016
26	1	26	28459	2	1017
27	1	27	29492	2	1033
28	1	28	30509	2	1017
29	1	29	31542	2	1033
30	1	30	32576	1	1034
31	1	31	33592	2	1016
			Averages:	1	1020

6. Review the column “Rel. Time” (Relative Time). This is the time, in milliseconds, from the beginning of the experiment to each stm+ event.

7. Compare the “Rel. Time” column to the “Elapsed Time” column. Elapsed Time is the time duration, in milliseconds, of each trial. The Elapsed Time should

equal the difference between the Relative Time of the current and previous trials. In this experiment, that would be roughly 1000ms.

7. Most importantly, review the “Offset” column. The Offset value listed for each trial is the number of milliseconds between the stim+ and DIN events in Net Station.

The Offset value represents the difference between the timestamp E-Prime assigned to the stim+ event it sent to Net Station, and the timestamp Net Station assigned the DIN event it created when the AV Tester detected the flash of the Stimulus rectangle.

8. Since E-Prime and Net Station both get time from the amplifier clock, the Offset for all trials should not vary by more than a millisecond or two for the entire recording. Look carefully through the ETT table for your test file and verify that this is true.

How to review a square wave file

If you used the Neurotravel amplifier for your testing and recorded a square wave signal in your file, you will need to review the results in Net Station.

1. Launch Net Station on the Macintosh and Click “Select” to open your timing test recording file.

2. Expand the “Events” track and select a “stim+” event to view its timestamp and duration.

2. Locate the nearest square wave, and drag-select the section of the waveform between the falling edge and the stim+ event. The duration of this selection is the “offset”.

Limitations of the Single Clock

If the amplifier clock was perfectly precise and data transmission was instantaneous, there would be absolutely no drift in the Offset reported by the Event Timing Tester application and viewed in your timing test recording. However, our testing has shown that “micro-drifts” occur over time. Looking at the drifts that occur during numerous 10 minute recordings, the average drift is approximately zero, with a maximum drift of approximately plus or minus 0.75 milliseconds and a standard deviation of approximately plus or minus 0.3 milliseconds. A one-hour recording will contain many micro-drifts in both the positive and negative direction. These negative and positive micro-drifts largely

cancel each other out, with a resulting total drift of up to plus or minus 1 millisecond.

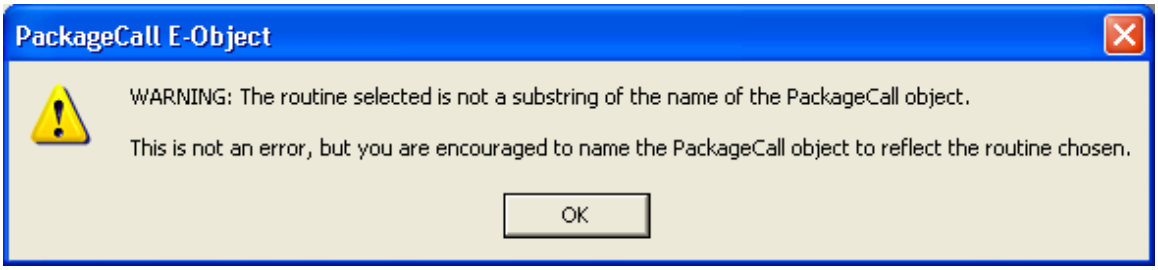
Net Station Package Call Reference

This section details the set of routines that are available within the Net Station Package File and how to properly call them.

PackageCall Method

Routines contained within a package file are typically called by dragging a PackageCall object from the E-Studio Toolbox, and dropping it into a Procedure object at the location within the experiment where the call is to be invoked.

When using the PackageCall object it is strongly recommended that you rename the object to reflect the specific Package File and Routine that are being referenced within the object. Newer version of E-Prime will issue the following warning if you have not followed this recommendation.


PackageCall E-Object
WARNING: The routine selected is not a substring of the name of the PackageCall object.
This is not an error, but you are encouraged to name the PackageCall object to reflect the routine chosen.
OK

It is also a common convention, when calling routines within the Net Station package file, to create a name for each object by abbreviating “Net Station” as “NS” and then concatenating on the name of the routine being called. For example, a PackageCall object that is configured to call the “NetStation_Init” routine would be named “NSInit”.

Inline Script Method

It is also possible to directly call a package file routine by writing E-Basic script within an Inline object that is placed at the desired location within the experiment. However, unless there is a need to make a series of package calls in sequence, or mingle package calls with additional E-Basic script, this calling method is not typically recommended.

If you are making direct package calls via E-Basic script, please refer to the following reference pages which document each Net Station package routine and provide examples of the script commonly required to successfully invoke each routine.

NetStation_Init

Description

Initializes the Net Station system. This routine attempts to configure the Net Station Package File and ready it for use. It will verify that the required Startup Info has been collected (i.e. Age, Sex, Handedness, etc.) and attempt to establish communications with Net Station. If valid communications are established, recording will be initiated and Net Station and E-Prime will be synchronized. The specified Cell List will then be processed and information about each cell used in the experiment will be sent to Net Station. After this information is sent to Net Station, recording will be then be stopped again (unless the optional bStopRecording parameter is explicitly specified as False - see Parameters below).

Syntax

```
NetStation_Init c, strState, objCellList, strCommMethod,
            strCommMethodParams [, bStopRecording]
```

Default Parameters

```
c, "on", CellList
```

Parameters

c As Context

The current experiment context. The routines in the Net Station Package File may optionally use the experiment context to retrieve the current values of attributes stored in the context. The context is typically the first parameter of every package file routine.

strState As String

The requested state of Net Station communications (“on”, "off"). When “on” is specified the system will attempt to establish communications with Net Station and a non-fatal error will be produced if the attempt fails. When “off” is specified the system will not attempt to establish communications with Net Station and no errors will be generated.

objCellList As List

The Cell List to use for this experiment. Specifies a List object which contains attributes (minimally CellNumber and CellLabel) which describe and identify the cells that may be used within the experiment.

strCommMethod As String

The requested Net Station communications method ("socket", "serial”). If not specified the communications method parameters will be read from the Method setting in the NETSTATION.INI file.

strCommMethodParams As String

A comma delimited string of parameters associated with the Net Station communications method. If not specified the communications method parameters will be read from the MethodParams setting in the NETSTATION.INI file.

When strCommMethod="socket", use this parameter to specify the IP address and port of the remote Net Station computer, e.g. "168.254.100.1". If no port number is specified the default port expected by Net Station will be used. If you want to use a different port number then append a colon and the port number to the IP address string, e.g. "168.254.100.1:55513".

When strCommMethod="serial", use this parameter to specify the id of the serial port you want to use for serial communications, e.g. "1"

Optional bStopRecording As Boolean

Indicates whether or not Net Station should stop recording immediately after being initialized. If the parameter is not specified it will default to True.

Example Call(s)

```
‘ Initialize the system for active socket communications on IP
‘ address 11.0.0.42
NetStation_Init c, "on", CellList, "socket", "11.0.0.421"

‘ Turn off active socket communications on IP address 11.0.0.42.
‘ Useful when testing Net Station experiments on a lab experiment
‘ development machine that is not connected to the Net Station
‘ hardware.
NetStation_Init c, "off", CellList, "socket", "11.0.0.421"

‘ Initialize the system using the communication parameters specified in
‘ the NETSTATION.INI file.
NetStation_Init c, "on", CellList

‘ Initialize the system for active serial communications on COM 1
NetStation_Init c, "on", CellList, "serial", "1"

‘ Initialize the system for active serial communications on COM 2.
‘ Do not stop recording after initialization
NetStation_Init c, "on", CellList, "serial", "2", False
```

NetStation_SendRespEvent

Description

Sends a RESP event to the Net Station system. Net Station requires that it be notified of any participant response of interest that was collected during each trial. This routine obtains the relevant response information from an object that is passed as the second parameter. This parameter must be a reference to the object within the trial sequence that was responsible for collecting the response.

It is important to verify that the allowable response interval within each trial has fully completed before calling this routine. For example, if PreRelease is in use by objects contained within the trial sequence and a participant responds near the very end of the response interval there is the potential for invalid or incomplete response data to be sent to Net Station (see *E-Prime User's Guide, Chapter 3: Critical Timing, Section 3.3.1.1* for details on how to avoid this problem.)

If the routine determines that there was no response collected then the routine will return immediately and no response related information will be sent to Net Station.

Syntax

```
NetStation_SendRespEvent c, objRteObject [, strEventID]
```

Default Parameters

```
c,
```

Parameters

c As Context

The current experiment context. The routines in the Net Station Package File may optionally use the experiment context to retrieve the current values of attributes stored in the context. The context is typically the first parameter of every package file routine.

objRteObject As RteObject

A reference to either the stimulus object that collected the response or a ResponseData object. ResponseData objects are obtained programmatically from the Responses collection that is a property of each InputMask defined for the stimulus object (see InputMask.Responses in the
E-Basic Help for more information).

Optional strEventID As String

An optional four character code to be used associated with the response event. If the parameter is specified its value must be exactly four characters in length or a runtime error will be generated. If the parameter is not specified it will default to 'resp'.

Example Call(s)

```
‘ Send response information collected by an object named Stimulus
NetStation_SendRespEvent c, Stimulus

‘ Send response information collected by an object named Stimulus but
‘ use “rsp1” as the id for the response
NetStation_SendRespEvent c, Stimulus, “rsp1”

‘ Send response information for each of several responses collected by
‘ an object named Stimulus and assign a unique id to each response.
Dim n As Long
For n = 1 To Stimulus.InputMasks(1).Responses.Count
```

```
      NetStation_SendRespEvent c, Stimulus.InputMasks(1).Responses(n),_
            "rsp" & n
Next n
```

NetStation_SendTrialEvent

Description

Sends a Trial event to the Net Station system. Prior to the end of each trial, Net Station requires that it be sent all relevant timing information related to any critical stimulus presentation or object/event of interest that occurred during the trial.

The routine examines the specified object's "Tag" property to obtain a unique ID for the current trial event. The object's Tag property must be set to a unique four character code. If the ID code is not exactly four characters in length a runtime error will be generated.

The routine includes two optional parameters that can be used to associate additional user defined data with the current trial event.

Syntax

```
NetStation_SendTrialEvent c, objTrialEvent [, nKeyValue] [, strKeyName]
```

Default Parameters

```
c,
```

Parameters

c As Context

The current experiment context. The routines in the Net Station Package File may optionally use the experiment context to retrieve the current values of attributes stored in the context. The context is typically the first parameter of every package file routine.

objTrialEvent As RteRunnableObject

A reference to an object in the trial sequence. The object's Tag property must be set to a unique four character code or a runtime error will be generated.

Optional nKeyValue As Integer

Optional user defined data that is to be associated with the trial event. If a value is specified it must be numeric (integer). If a value is not specified it will default to 0.

Optional strKeyName As String

Optional user defined four character code to associate with the nKeyValue. If the parameter is specified its value must be exactly four characters in length or a runtime error will be generated. If the parameter is not specified it will default to 'argu'.

Example Call(s)

```
' Send trial event information for the object that presents the
' fixation
NetStation_SendTrialEvent c, Fixation

' Send trial event information for the object that presents the
' critical stimulus and send an addition user defined key and data
' obtained from an attribute that exists in the experiment context
NetStation_SendTrialEvent c, Stimulus, c.GetAttrib("data"), "data"
```

NetStation_SendTRSPEvent

Description
Sends a TRial SPecific (TRSP) event to the Net Station system. Net Station requires that a TRSP event be sent prior to the end of each trial so that it is notified of the behavioral data related to the critical stimulus and response.

The routine includes an optional parameter that may be used to specify a reference to a user defined List object within the experiment which has been configured to describe additional key data that should be associated with the current trial. If a List object is referenced, it must be specified to minimally contain the named attributes of KeyID, KeyName and KeyDataType. The routine will process the KeyList by retrieving each KeyName, looking up its current value in the experiment Context, and then appending all of the key information to the end of the TRSP event.

Syntax
```
NetStation_SendTRSPEvent c, objStimulus [, objKeyList]
```

Default Parameters
```
c,
```

Parameters
c As Context

The current experiment context. The routines in the Net Station Package File may optionally use the experiment context to retrieve the current values of attributes stored in the context. The context is typically the first parameter of every package file routine.

objStimulus As RteRunnableInputObject

A reference to the object in the trial sequence that is responsible for presenting the critical stimulus and optionally collecting the response data. The specified object's Tag property must be set to a unique four character code within the trial sequence or a runtime error will be generated.

Optional objKeyList As List

An optional Key List to use for this trial. Specifies a List object that contains attributes which describe a set of user defined keys that are to be sent to Net Station and associated with the current trial. If the parameter is not specified then no additional keys will be sent to Net Station. If the parameter is specified, the KeyList object is expected to minimally contain the named attributes of KeyID, KeyName and KeyDataType. If a specified KeyName cannot be found in the context a runtime error will be generated. The value of the KeyName attribute must a string. The value of the KeyID attribute must be a four character code. The value of the KeyDataType attribute must be one of the following codes: "shor", "long", "bool", "TEXT", "sing", "doub", "type". If the value of KeyDataType is left blank it will default to "shor". Please refer to your Net Station documentation for more information on valid key values and ranges for the different data types.

Example Call(s)
```
‘ Send the TRSP event assuming that the object named “Stimulus” is
‘ responsible for presenting the critical stimulus and collecting
‘ the response.
NetStation_SendTRSPEvent c, Stimulus

‘ Send the TRSP event as above but also include the additional
‘ user defined keys that are listed in a List object named KeyList
NetStation_SendTRSPEvent c, Stimulus, KeyList
```

NetStation_SendStartRecording

Description
Start the Net Station system recording. Recording is typically started at the beginning the experiment or after any extended rest or break periods within the execution of the experiment have occured.

Syntax
```
NetStation_StartRecording c
```

Default Parameters
c

Parameters
c As Context

The current experiment context. The routines in the Net Station Package File may optionally use the experiment context to retrieve the current values of attributes stored in the context. The context is typically the first parameter of every package file routine.

Example Call(s)
```
‘ Start the Net Station system recording.
NetStation_StartRecording c
```

NetStation_StopRecording

Description
Stop the Net Station system recording. Recording is typically stopped at the end of a block of trials or immediately prior to any extended rest or break periods that may occur within the experiment.

Syntax
```
NetStation_StopRecording c
```

Default Parameters
c

Parameters
c As Context

The current experiment context. The routines in the Net Station Package File may optionally use the experiment context to retrieve the current values of attributes stored in the context. The context is typically the first parameter of every package file routine.

Example Call(s)
```
‘ Stop the Net Station system recording
NetStation_StopRecording c
```

NetStation_TrialBegin

Description

Notifies the Net Station system of the start of a new trial. This call is typically executed at the very beginning of each trial procedure or after the presentation of any trial level instructions which may require input from the participant in order to begin the critical stimulus presentation sequence.

When this routine is called, it will determine the currently active cell by searching the experiment context for an attribute named “CellNumber”. Once the current cell is determined the routine uses this information to increment the number of observation for the current cell as well as the current Net Station trial number.

Syntax

```
NetStation_TrialBegin c
```

Default Parameters

c

Parameters

c As Context

The current experiment context. The routines in the Net Station Package File may optionally use the experiment context to retrieve the current values of attributes stored in the context. The context is typically the first parameter of every package file routine.

Example Call(s)

```
‘ Notify Net Station of the start of the current trial
NetStation_TrialBegin c
```

NetStation_TrialEnd

Description
Notifies the Net Station system of the end of the current trial and verifies that the message/communication buffers are cleared so that the system is prepared for the next trial. This call is typically executed at the very end of each trial procedure.

Syntax
```
NetStation_TrialEnd c
```

Default Parameters
c

Parameters
c As Context

The current experiment context. The routines in the Net Station Package File may optionally use the experiment context to retrieve the current values of attributes stored in the context. The context is typically the first parameter of every package file routine.

Example Call(s)
```
‘ Notify Net Station of the end of the trial
NetStation_TrialEnd c
```

NetStation_Uninit

Description
Uninitializes the Net Station system. This call is typically executed at the end of the experiment (e.g. at the end of the session procedure) and is responsible for coordinating the termination of communications with Net Station.

Syntax
```
NetStation_Uninit c
```

Default Parameters
```
c
```

Parameters
c As Context

The current experiment context. The routines in the Net Station Package File may optionally use the experiment context to retrieve the current values of attributes stored in the context. The context is typically the first parameter of every package file routine.

Example Call(s)
```
‘ Uninitialize the Net Station system
NetStation_Uninit c
```